ALGORITMO DE SELECCIÓN Y VALIDACIÓN DEL MÉTODO DE CLUSTERIZACIÓN ÓPTIMO PARA DATOS NO SUPERVISADOS

Luisa Fernanda Pastrán Ramírez
Santiago Gongora Aya

Universidad Tecnológica de Pereira
Facultad de Ingeniería Industrial
Maestría en Investigación de Operaciones y Estadística
Bogotá D.C
2021

ALGORITMO DE SELECCIÓN Y VALIDACIÓN DEL MÉTODO DE CLUSTERIZACIÓN ÓPTIMO PARA DATOS NO SUPERVISADOS

Trabajo de grado para optar al título de
Magister en Investigación de Operaciones y Estadística

Luisa Fernanda Pastrán Ramírez.
Santiago Gongora Aya.

Director:
Milton Januario Rueda
Doctor en Estadística

Universidad Tecnológica de Pereira
Facultad de Ingenieria Industrial
Maestría en Investigación de Operaciones y Estadística
Bogotá D.C
2021

ÍNDICE

Índice de figuras

Índice de tablas

Agradecimientos

Todo el esfuerzo y dedicación e para Dios quien me llenó de sabiduría, inteligencia y paciencia.

A mis padres Fermín y Amparo por darme amor, ánimo y fuerza. Por inculcarme amor, ganas de luchar y demás valores en el transcurso de mi vida.

A ti hombre maravilloso, que siempre ha confiado en mí y siempre ha tenido palabras de aliento.

Luisa Fernanda Pastrán Ramirez

Este trabajo lo dedico especialmente a Dios, que es el verdadero propulsor y motor de mi vida.
A mis papás Obed y Nubia por su continuo amor, paciencia, sabiduría, dedicación y ejemplo de vida. A mis hermanos Felipe y Laura por su incondicional apoyo y soporte en esta vida.
Para una persona muy especial que llegó a cambiar mi vida; a mi amiga incondicional y amada esposa, Keila, por ser el impulso que me da fuerzas para seguir adelante cada día y hacerme la vida más feliz, ya que creyó en mí, desde el primer momento.
Para todas las personas que amo y que siempre están conmigo en los buenos y los malos momentos, mi familia y amigos.

Santiago Góngora Aya

También queremos agradecer de manera especial a Milton Januario Rueda por impulsarnos a dar lo mejor, al aceptar ser nuestro director de tesis. Por su amistad, sabiduría, paciencia y entrega por lo que hace, nos ha dejado un profundo deseo por la investigación estadística.

Autores

Resumen

Este trabajo de investigación analiza las diferentes generalidades de clusterización en la cual se estudiarán diferentes técnicas de clasificación como: aprendizajes Supervisado y no supervisado partiendo desde el proceso KDD (Knowledge Discovery in Databases) el cual es interactivo e iterativo que involucra numerosos pasos con la intervención del usuario en la toma de decisiones.

Continuará luego con los métodos de segmentación clasificados en 3 grandes grupos, que son: partición, densidad y difusos, conociendo sus usos, beneficios, restricciones y las posibles mejoras realizadas.

Estos métodos de segmentación de datos se encargan de hacer un análisis de perfilamiento de los datos, con el objetivo de encontrar patrones y entender su comportamiento, por este motivo es necesario realizar una validación de índices para los clústeres.

Por lo tanto en este trabajo de investigación se propone un algoritmo que permite interactuar diferentes métodos de segmentación para datos "no supervisados", evaluando los resultados de las agrupaciones mediante los criterios de índices de validación, que permiten evidenciar la eficacia de la segmentación de cada método, comparándolas entre si, validando su comportamiento a través de un conjunto k de segmentaciones.

Este Algoritmo identifica el conjunto de agrupamiento óptimo al segmentar en un cuadro característico para luego encontrar el mejor método de segmentación.

Introducción

En la actualidad la información utilizada para analítica y minería de datos es, en su gran mayoría, de carácter número, por lo tanto los métodos de segmentación (Clusterización) constituyen un enfoque de investigación, en esta área del conocimiento y son utilizados en distintos campos como una importante fuente de investigación, para entender el comportamiento de los individuos, haciendo un estudio de perfilamiento de los datos para la toma de decisiones.

Uno de los inconvenientes más comunes, es la falta de validación de los resultados de los conglomerados(clústeres) y esto conlleva en muchos casos a una mala interpretación del comportamiento de los individuos de estudio. Por ende, las decisiones que se tomen teniendo como base las conclusiones de estos estudios, no necesariamente son las más adecuadas y por lo tanto pueden generan resultados poco robustos.
Esto se da muy a menudo, ya que no se evalúa el método de segmentación utilizado, ni se comparan los resultados con otro algoritmo de clusterización.

En el primer capítulo se encuentra información referente a las diferentes técnicas más estudiadas y conocidas de clasificación y utilizando el método KDD (Knowledge Disco-very in Databases) para el desarrollo del trabajo.

En el segundo capítulo se estudiarán algoritmos que permitan clusterizar individuos teniendo una matriz de información no jerárquica para variables numéricas. En el tercer capítulo se estudiarán los indices de validación internos, relativos y externos para con ellos definir cuales se tendrán en cuenta en nuestro último capítulo.

En el cuarto capítulo la presente investigación se enfocará en estudiar y encontrar las metodologías de segmentación más usadas para datos no supervisados, estableciendo un esquema de evaluación para los resultados con criterios de validación interna y sugiriendo algunos índices de validación externa.

Así, el presente trabajo permitirá mostrar el escenario más adecuado para cada método optimizando resultados, brindando la calidad y robustez esperada, para la adecuada toma de decisiones.

Objetivos

Objetivo General

Diseñar un algoritmo el cual permita identificar por medio de la distribución de los datos, el método de segmentación óptimo para un conjunto de datos no supervisado, teniendo como referencia índices de validación interna.

Objetivos Específicos

1. Analizar diferentes métodos de segmentación teniendo como enfoque los requerimientos propios y su aplicabilidad.
2. Estudiar los índices de validación interna más utilizados en la validación de los resultados de los conglomerados.
3. Diseñar y programar un algoritmo que identifique la metodología óptima de clasificación utilizando el software R.

Capítulo I
GENERALIDADES DE CLUSTERIZACIÓN

El constante crecimiento de la tecnología ha permitido recopilar de manera más eficiente información para bases de datos. Por tal motivo existe la necesidad de entender el comportamiento de los componentes de dichas tablas, bases de datos (DB por sus siglas en inglés), "data frames" que aporten referencias significativas en la toma de decisiones respecto a los individos de estudio. Para ello, el análisis de conglomerados o clúster, busca agrupar elementos o variables, con el objetivo de lograr la máxima homogeneidad en cada grupo y heterogeneidad entre los grupos generados. Este análisis fundamentalmente se conoce como una técnica exploratoria o descriptiva pero no explicativa, sin embargo su valor tanto teórico como práctico es fácilmente evidenciable y constituye una etapa primordial en cualquier análisis, además de ser la base para desarrollos analíticos en cualquier conjunto de datos.

Existen diferentes procedimientos para construir dichos grupos y diferentes formas de determinar cómo se mide la similitud. Para esto se tiene en cuenta la distancia entre las observaciones, que a su vez están determinadas por el tipo de variables que se están analizando, sean cuantitativas, cualitativas ordinales (al resultado se le puede asignar un número cuyo orden tiene sentido, pero no la diferencia entre dos valores) y las cualitativas nominales (que corresponden a una etiqueta y donde la similitud se determina como simple coincidencia de valores).

Las técnicas más estudiadas y conocidas de clasificación las expone (Russell y Norvig, 2002) que son: *aprendizaje por instrucción*, *aprendizaje por deducción*, *aprendizaje por inducción*. Del último hay dos tipos principales:

1. **Aprendizaje con ejemplos o "Supervisado":** Se entiende como un algoritmo que produce una función para establecer una correspondencia entre las entradas y las salidas deseadas del sistema. Por tanto, los algoritmos de regresión cuando el resultado a predecir es un atributo numérico y los algoritmos de clasificación cuando el resultado a predecir es un atributo categórico.

2. **Aprendizaje por observación y descubrimiento o "no Supervisado":** Es cuando todo el proceso de modelamiento se lleva a cabo sobre un conjunto de ejemplos formado tan sólo por entradas al sistema, es decir, sus rasgos. No se tiene información sobre las categorías o clasificación de esos ejemplos. Por ende constituye un tipo de aprendizaje por observación y descubrimiento, donde el sistema de aprendizaje analiza una serie de

entidades y determina si algunas tienen características comunes, por lo que pueden ser agrupadas.

El descubrimiento del conocimiento en bases de datos (KDD, del inglés Knowledge Discovery in Databases) es básicamente un proceso automático en el que se combina descubrimiento y análisis. El proceso consiste en extraer patrones en forma de reglas o funciones, a partir de los datos, con el fin de que el usuario los pueda analizar. Esta tarea implica generalmente preprocesar los datos, hacer minería de datos (data mining) y presentar resultados (Agrawal, Srikant, y cols., 1994) (Chen, Han, y Prinn, 2006) (Han y Kamber, 2001).

El proceso KDD que se muestra en la figura 1.1 (Timarán Pereira S.R, 2016) es interactivo e iterativo; involucra numerosos pasos con la intervención del usuario en la toma de decisiones. Se resume en las siguientes etapas:

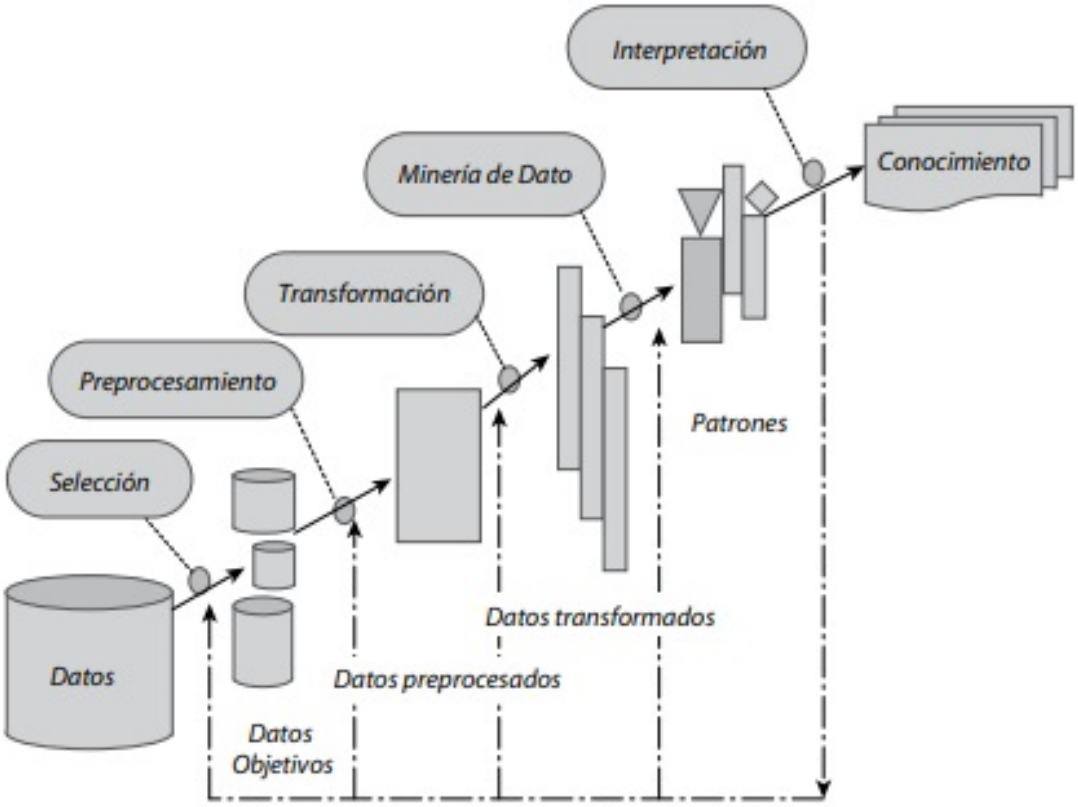

Figura 1.1: Proceso KDD

A continuación se presenta la descripción de cada etapa del proceso:

- **Selección:** En la etapa de selección, una vez identificado el conocimiento relevante, prioritario y definidas las metas del proceso KDD, desde el punto de vista del usuario final, se debe crear un conjunto de datos objetivo, seleccionando todo el conjunto de datos o una muestra representativa de este, sobre el cual se realiza el proceso de descubrimiento. La selección de los datos varía de acuerdo con los objetivos del negocio.

- **Pre-procesamiento/limpieza:** En la etapa de pre-procesamiento/limpieza (data cleaning) se debe analizar la calidad de los datos, aplicar operaciones básicas como la remoción de datos ruidosos, seleccionar estrategias para el manejo de datos desconocidos (missing y empty), datos nulos y datos duplicados como técnicas estadísticas para su reemplazo. En esta etapa, es de suma importancia la interacción con el usuario y/o analista.
  Los datos ruidosos (noisy data) son valores que están significativamente fuera del rango de valores esperados; se deben principalmente a errores humanos, cambios en el sistema, información no disponible a tiempo y a fuentes heterogéneas de datos. Los datos desconocidos *empty* son aquellos a los cuales no les corresponde un valor en el mundo real y los *missing* son aquellos que tienen un valor que no fue capturado. Los datos nulos, son datos desconocidos que son permitidos por los sistemas gestores de matrices de información relacionales (sgbdr). En el proceso de limpieza todos estos valores se ignoran, se reemplazan por un valor por omisión, o por el valor más cercano, es decir, se usan métricas de tipo estadístico como media, moda, mínimo y máximo para reemplazarlos.

- **Transformación/reducción:** En la etapa de transformación/reducción de datos, se buscan características útiles para representar los datos dependiendo de la meta del proceso. Se utilizan métodos de reducción de dimensiones o de transformación para disminuir el número efectivo de variables bajo consideración o para encontrar representaciones invariantes de los datos (Fayyad, Piatetsky-Shapiro, y Smyth, 1996).

  (Rueda, Moya, y Silva, 2011) muestra una de las técnicas, *el análisis factorial* el cual se aplica normalmente para el análisis de variables continuas (numéricas) con dos objetivos específicos:

  - El primero de ellos es descripción o comprobación.
  - El segundo, reducir el tamaño de un gran número de variables obteniendo nuevos factores ortogonales, que por ser combinación lineal de las variables originales recogen una gran proporción de la información suministrada. En el caso de variables categóricas, la técnica apropiada, es el análisis de correspondencias múltiples y busca utilizar la información de los factores determinados, mediante el análisis de correspondencias, con el fin de perfilar comportamientos de las variables a analizar para luego realizar una segmentación natural utilizando como insumo los factores determinados por el análisis factorial.

- **Minería de datos (data mining):** (Timarán Pereira S.R, 2016) El objetivo de la etapa minería de datos, es la búsqueda y descubrimiento de patrones insospechados y de interés, aplicando tareas de descubrimiento como clasificación (Quinlan, 1986) (Wang, Iyer, y Vitter, 1998) , segmentación (Ng y Han, 1994), (Zhang, Ramakrishnan, y Livny, 1996),

patrones secuenciales (Agrawal y Srikant, 1995) y asociaciones (Agrawal y cols., 1994), entre otras.
Las técnicas de minería de datos crean modelos que son predictivos y/o descriptivos.

- **Interpretación/evaluación:** (Timarán Pereira S.R, 2016) En la etapa de interpretación/evaluación, se deben analizar los patrones descubiertos y posiblemente se retorna a las anteriores etapas para posteriores iteraciones. Esta etapa puede incluir la visualización de los patrones extraídos, la remoción de los patrones redundantes o irrelevantes y la traducción de los patrones útiles en términos que sean entendibles para el usuario.

Dentro de la minería de datos se encuentran diferentes tipos de tareas, las cuales pueden considerarse como un tipo de problema para ser resuelto por un algoritmo de minería de datos. (Hernández, Ramírez, y Ferri, 2005).

De las muchas tareas de minería de datos la presente investigación se enfocará, en una de ellas, que es la segmentación (clustering) aplicado en métodos jerárquicos y no jerárquicos.

Entre los métodos no Jerárquicos uno de los algoritmos más conocidos y utilizado para variables numéricas es el *k-means*, el cual agrupa y tiene como objetivo la partición de un conjunto de n individuos en K grupos en el que cada observación pertenece al grupo más cercano de la media del mismo. Las nubes dinámicas difuso juegan un papel importante en este método ya que son una generalización de este principio y vez de asignar un objeto a una sola clase, lo que se hace es asignar el objeto a varias las clases en cierto porcentaje cada partición.

Benzecrí (J.-P. Benzécri, 1977) (J. Benzécri, 1984) y Lebart (Lebart, 1994) profundizaron en la clasifiación de variables categóricas ya que implementaron un método de segmentación para estas variables teniendo en cuenta coordenadas. Tiempo después se notó que haciendo algunos cambios en el algoritmo de *k-means* se obtiene el algoritmo *k-moda*, el cual al hacer cambios en la disimilitud y algunas sustituciones ayudarán a la clasificación de variables categóricas (Pastrán y Roa, 2015).

El modelo de clasificación basado en árboles de decisión es probablemente el más utilizado y popular entre los módelos Jerárquicos por su simplicidad y facilidad para entender. (Han y Kamber, 2001), (Sattler y Dunemann, 2001). Este modelo tiene su origen en los estudios de aprendizaje de máquina y es un método de aprendizaje supervisado que construye árboles de decisión a partir de un conjunto de casos o ejemplos denominados *conjunto de entrenamiento* (training set) extraídos de la base de datos. También se escoge un conjunto de prueba (text set), cuyas características son conocidas, con el fin de evaluar el árbol.

En la literatura se encuentran más algoritmos (métodos) de segmentación en los cuales se profundizarán en el próximo capítulo enfocados especificamente en métodos para datos no supervisados.

Luego de estudiar estos algoritmos se estará analizando validaciones que se ejecutarán en los métodos seleccionados en las matrices de información de estudio para evaluar si son efectivos para entregar buenos resultados. El procedimiento de evaluación de los resultados de un

algoritmo de agrupamiento se conoce bajo el término *validez del clúster*. Esto con el objetivo de conocer, profundizar y enfatizar que este proceso de calidad sea tenido en cuenta antes de dar por hecho el resultado de estos mismos.

En términos generales, hay tres enfoques para investigar la validez del clúster (Theodoridis y Koutroumbas, 1999):

- El primero se basa en *criterios externos*. Esto implica que se evalúan los resultados de un algoritmo de agrupamiento basado en una estructura pre-especificada, que se impone en un conjunto de datos y refleja la intuición acerca de la estructura de la agrupación del conjunto de datos.
- El segundo enfoque se basa en *criterios internos*. Se puede evaluar el resultados de un algoritmo de agrupación en términos de cantidades que involucran los vectores de los datos que establecen ellos mismos (por ejemplo, matriz de proximidad).
- El tercer enfoque de la validez de la agrupación se basa en *criterios relativos*. Aquí la idea básica, es la evaluación de una estructura de agrupamiento comparando a otros esquemas de agrupación resultantes con el mismo algoritmo, pero con parámetros diferentes.

El capítulo 3 estará analizando estos índices de validación profundizando específicamente en el segundo enfoque anteriormente mencionado (validación interna) ya que son los más utilizados en la actualidad.

Para el capítulo 4 se elaborará un algoritmo que permita mostrar los resultados obtenidos de acuerdo al análisis realizado a un conjunto de datos no supervisado, garantizando cúal es el mejor método para el conjunto de datos.

Capítulo II
MÉTODOS DE SEGMENTACIÓN PARA DATOS NUMÉRICOS

La importancia de tener un perfilamiento de los individuos en que son objeto estudio ha generado la necesidad en la minería de datos de implementar algoritmos de segmentación que nos permiten entender el comportamiento de la información suministrada por cada individuo.

Por lo tanto, en este capítulo se estudiarán algoritmos que permitan clusterizar individuos teniendo algoritmos no jerárquicos para variables numéricas.

Estos métodos de segmentación se resumen en 3 grupos:

- Métodos de partición
- Algoritmos de densidad
- Métodos difusos

2.1. Métodos de Partición

El propósito de estos métodos es proporcionar una partición convencional de m objetos en k clusters (clases) disjuntos. En la literatura, a menudo se les conoce como partición dura o nítida, a diferencia de la agrupación difusa. Por definición, un objeto solo puede pertenecer a un único grupo y cada grupo debe tener al menos un elemento (de lo contrario, habría menos de k clústeres).

El algoritmo clasificatorio suele ser iterativo, es decir, una partición inicial se mejora paso a paso en el análisis hasta que no se puede lograr una mejora sustancial. La definición de una partición inicial requiere una especificación a priori del número de clases.

Suponga que la "bondad" de la partición se mide por la función J, cuyo valor se debe disminuir tanto como sea posible para lograr una mayor optimización del resultado. Entonces, un algoritmo de partición muy general implica los siguientes pasos (Hartigan, 1985):

1. Especificar una partición inicial en k clústeres y calcule el valor de J.

2. Cambiar la partición para disminuir el valor de J tanto como sea posible, sin modificar k (es decir, vacíos o nuevos clústeres no pueden aparecer como resultado de este cambio).

3. Si no es posible una reducción de J, el análisis se detiene con la partición real como resultado final. De lo contrario, se continúa las iteraciones en el paso 2.

Los diferentes procedimientos varían en la definición de la función de bondad J y en las operaciones permitidas en el paso 2 para modificar la partición real. Una propiedad fundamental del anterior algoritmo general es que el resultado a menudo puede ser solo un óptimo local, es decir, no necesariamente es la mejor clasificación de los objetos dados en k grupos de acuerdo con el criterio J.

Es muy posible que desde una partición inicial diferente, se pueda alcanzar un valor aún más pequeño de J. Por la misma razón, el análisis también puede quedar "atrapado" en soluciones "muy malas", es decir, no adecuadas. Este problema se puede eludir al realizar las iteraciones de decenas de particiones iniciales diferentes y reteniendo el "mejor" resultado. Nunca se puede estar 100 % seguros de haber alcanzado el óptimo absoluto (global). Para obtener esto, se deben verificar todas las particiones posibles, que es una tarea imposible de alcanzar para valores grandes de m.

La partición se puede modificar en el Paso 2 de dos maneras:

1. Se examina cada objeto por separado, cómo su reubicación del grupo real a otro grupo influye en el valor de J. Los objetos que causan una disminución de J se reubican en el grupo para el cual esta disminución es el máximo. Es posible que muchos o incluso todos los objetos deban ser reubicados en un solo paso, y solo se espera que el nuevo valor resultante de J sea más pequeño que el anterior.

2. El objeto para el cual se logra la disminución máxima de J se selecciona y luego se traslada al nuevo grupo. Esta estrategia implica una disminución monótona de J, y es definitivamente más lenta que el método anterior.

En este trabajo se estudiarán tres métodos de este grupo, al enfocarse en el comportamiento de su algoritmo ya que han sido escogidos por ser los más utilizados y aplicados a diferentes situaciones, sin embargo, la metodología planteada es aplicable a cualquier método que se requiera.

- El método *K-means* (MacQueen y cols., 1967)
- El Método *K-medoides* (K-medoides) (ROUSSEEUW y KAUFMAN, 1987)
- El Método *CLARA* (Ng y Han, 1994)

2.1.1. K-Means

El algoritmo *K-means* resuelve un problema de optimización, siendo la función a optimizar (minimizar) la suma de las distancias cuadráticas de cada objeto al centroide de su clúster.

Definición

K-means es un algoritmo de clasificación no supervisada (clusterización) que agrupa objetos en k grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo o cluster. Se suele usar la distancia cuadrática.

Los objetos se representan con vectores reales de d dimensiones $(x_1, x_2, \ldots, x_d)$ y el algoritmo *K-means* construye k grupos donde se minimiza la suma de distancias de los objetos, dentro de cada grupo $S = (S_1, S_2, \ldots, S_k)$, a su centroide. El problema se puede formular de la siguiente forma:

$$\min_{\mathbf{S}} E(\mu_i) = \min_{S} \sum_{i=1}^{k} \sum_{x_j \in S_i} \parallel x_j - \mu_i \parallel^2, \tag{2.0}$$

donde **S** es el conjunto de datos cuyos elementos son los objetos x_j representados por vectores, donde cada uno de sus elementos representa una característica o atributo. Se tendrán k grupos o clusters con su correspondiente centroide μ_i.

En cada actualización de los centroides, desde el punto de vista matemático, se impone la condición necesaria de extremo a la función $E(\mu_i)$ que, para la función cuadrática (2.0), sea

$$\frac{\partial E}{\partial \mu_i} = 0 \Longrightarrow \mu_i^{t+1} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \tag{2.1}$$

y se toma el promedio de los elementos de cada grupo como nuevo centroide.

Algoritmo

El algoritmo consta de los siguientes pasos:

1. *Inicialización:* Una vez se escoja el número de grupos, k, se establecen k centroides en el espacio de los datos, por ejemplo, escogiéndolos aleatoriamente.

2. *Asignación* de objetos a los centroides: Cada objeto de los datos es asignado a su centroide más cercano.

3. *Actualización* de centroides: Se actualiza la posición del centroide de cada grupo tomando como nuevo centroide la posición del promedio de los objetos pertenecientes a dicho grupo.

4. *S óptimo*: Se repiten los pasos 2 y 3 hasta que los centroides no se mueven, o se mueven por debajo de una distancia umbral en cada paso.

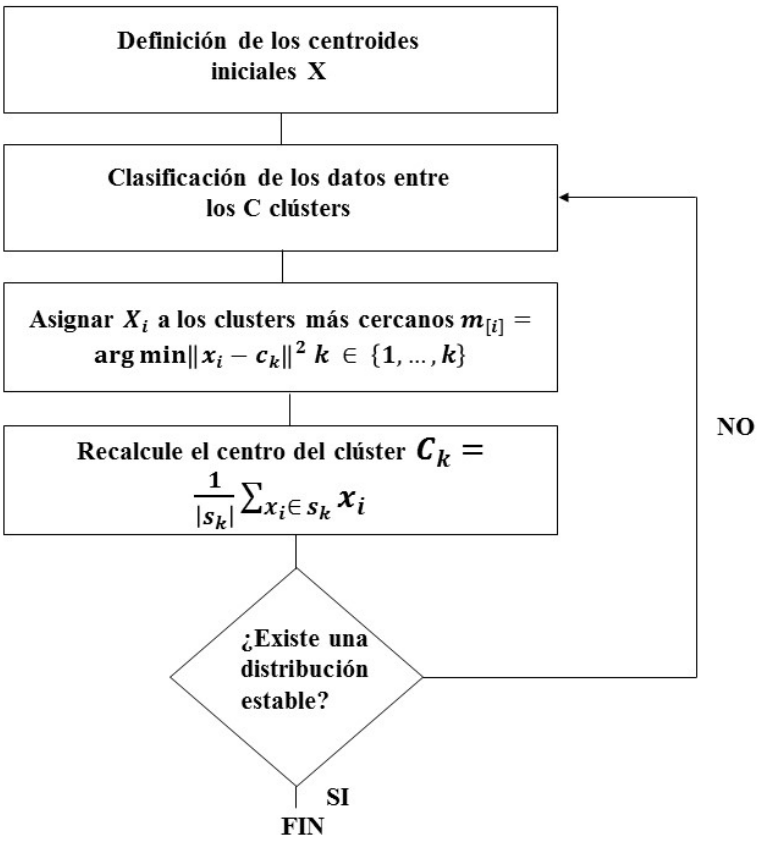

Figura 2.1: Algoritmo k-means

Usos y Beneficios

- La principal ventaja del algoritmo *K-means* es que es un método sencillo y rápido.
- El algoritmo *K-means* se adapta fá cilmente con heurísticas como el algoritmo de Lloyd ya que es fácil de implementar incluso para grandes conjuntos de datos. Por lo que ha sido ampliamente usado en muchas áreas como segmentación de mercados, visión por computadoras, geoestadística, astronomía y minería de datos en agricultura. También se usa como pre-procesamiento para otros algoritmos, por ejemplo para buscar una configuración inicial.

Restricciones

- El algoritmo sufre de algunas dificultades. *K-means* requiere varias iteraciones sobre todo el conjunto de datos, lo cual puede hacerlo muy costoso computacionalmente cuando se lo aplica a grandes bases de datos.
- El número de clústers K debe ser suministrado por el usuario.
- La búsqueda es propensa a quedar atrapada en mínimos locales.

Mejoras

Uno de los objetivos de los investigadores es buscar mejoras en los algoritmos ya que por temas computacionales o restricciones tienen falencias en algunos casos.

Como mejoras al algoritmo de *K-means*, se tiene lo siguiente:

- El algoritmo **EM** (Expectation Maximization) es un algoritmo que mejora el *K-means* ya que se puede adaptar para trabajar con datos categóricos en matrices de información mixtas.
  El algoritmo empieza especulando sobre los parámetros distribucionales y los usa para calcular las probabilidades de que cada objeto pertenezca a un clúster y usa esas probabilidades para re-estimar los parámetros de las mismas, hasta converger (se puede empezar especulando las probabilidades de que un objeto pertenezca a una clase).
  El cálculo de las probabilidades de las clases o los valores esperados de las clases es la parte de "expectation".
  El paso de calcular los valores de los parámetros de las distribuciones es "maximization", ya que maximiza la verosimilitud de las distribuciones, dados los datos.

- Se han creado heurísticas que mejoran la complejidad del algoritmo *K-means*, como por ejemplo el algoritmo **Honeycomb** ya que con el agrupamiento de instancias de muy alta dimensionalidad, se muestra que es factible mejorar el algoritmo *K-means* para la solución de instancias con un número alto de grupos. Concretamente, la heurística propuesta se enfoca en reducir la complejidad del algoritmo *K-means*.

2.1.2. K-medoides

El Algoritmo *K-medoides* es uno de los algoritmos más utilizados y pertenece a los métodos de partición. El objetivo del algoritmo es minimizar la disimilitud promedio de los objetos con respecto a su objeto seleccionado (centro) más cercano. De manera equivalente, se puede minimizar la suma de las diferencias entre el objeto y su objeto seleccionado al (centro) más cercano.

Definición

El método K-medoides (partición alrededor de los medoides) se define en encontrar una secuencia de objetos llamados "medoides" que se encuentren ubicados en el centro de cada grupo. Los objetos que se definen tentativamente como medoides se colocan en un conjunto S de objetos seleccionados. Si O es el conjunto total de objetos, el conjunto $U = O - S$ es el conjunto de objetos no seleccionados.

Algoritmo

El algoritmo se resume en dos fases:

1. **BUILD**: En la primera fase, se selecciona una colección de k objetos centroides para un conjunto inicial S
2. **SWAP**: En la segunda fase, se intenta mejorar la calidad del clúster intercambiando objetos seleccionados (centros) con objetos no seleccionados.

Al profundizar en el algoritmo, se tiene que para cada objeto p se mantienen dos números:

- Dp, la disimilitud entre p y el objeto más cercano en S
- Ep, la disimilitud entre p y el segundo objeto más cercano en S.

Estos números deben actualizarse cada vez que cambien los conjuntos S y U. Tener en cuenta que $D_j = E_j$, que p y S si y solo si $D_p = 0$.

La fase **BUILD** conlleva los siguientes pasos:

1. Se debe iniciar S agregando un objeto para el cual la suma de las distancias a todos los demás objetos sea mínima.
2. Luego se considera un objeto $i \in U$ como candidato para ser incluido en el conjunto de objetos seleccionado S.
3. Para un objeto $j \in (U - i)$ calcular D_j, que es la disimilitud entre j y el objeto más cercano en S.
4. Si $D_j > d(i, j)$, el objeto j contribuirá a la decisión de seleccionar el objeto i (porque la calidad de la agrupación puede beneficiarse); dejamos luego a $C_{ji} = max(D_j - d(j, i), 0)$.
5. Calcular la ganancia total obtenida sumando i a S como $g_i = \sum_{j \in U} C_{ji}$.
6. Elige ese objeto i que maximiza g_i; sea $\mathbf{S} = S \cup i$ y $\mathbf{U} = U - i$.

Estos pasos se realizan hasta que se hayan seleccionado k objetos.

La segunda fase, **SWAP**, intenta mejorar el conjunto de objetos seleccionados y, por lo tanto, mejorar la calidad del agrupamiento.

Esto se hace considerando todos los pares $(i, h) \in SxU$ y consiste en calcular el efecto T_{ih} en la suma de las disimilaridades entre los objetos y el objeto seleccionado más cercano causado por el intercambio de i y h, es decir, transfiriendo i de S a U y transfiriendo h de U a S.
El cálculo de T_{ih} implica el cálculo de la contribución K_{jih} de cada objeto $j \in U - h$ al intercambio de i y h. Tener en cuenta que $d(j, i) > D_j$ o $d(j, i) = D_j$.

1. K_{jih} se calcula teniendo en cuenta los siguientes casos:

a) si $d(j,i) > D_j$, entonces se producen dos subcasos:

   1) si $d(j,h) = D_j$, entonces $K_{jih} = 0$;
   2) si $d(j,h) < D_j$, entonces $K_{jih} = d(j,h) - D_j$.

   En ambos subcasos, $K_{jih} = min\,[d(j,h) - D_j, 0]$.

b) si $d(j,i) = D_j$, se tienen dos subcasos:

   1) Si $d(j,h) < E_j$, donde E_j es la disimilitud entre j y el segundo objeto seleccionado más cercano, luego K_{jih} = $d(j,h) - D_j$; nótese que K_{jih} puede ser positivo o negativo.
   2) Si $d(j,h) = E_j$, entonces $K_{jih} = E_j - D_j$; en este caso $K_{jih} > 0$.

En cada una de las subcasos anteriores tenemos:

$$K_{jih} = min[d(j,h), E_j - D_j] \tag{2.2}$$

2. Se Calcula el resultado total del SWAP como

$$T_{ih} = \sum K_{jih} | j \in U \tag{2.3}$$

3. Luego seleccionar un par $(i,h) \in S * U$ que minimice T_{ih}.

4. Si $T_{ih} < 0$ se realiza el intercambio, D_p y E_p se actualizan para cada objeto p, y se vuelve al Paso 1. Si min $T_{ih} > 0$, el valor del objetivo no puede disminuirse y el algoritmo se detiene. Por supuesto, esto sucede cuando todos los valores de T_{ih} son positivos y esta es precisamente la condición de detención del algoritmo.

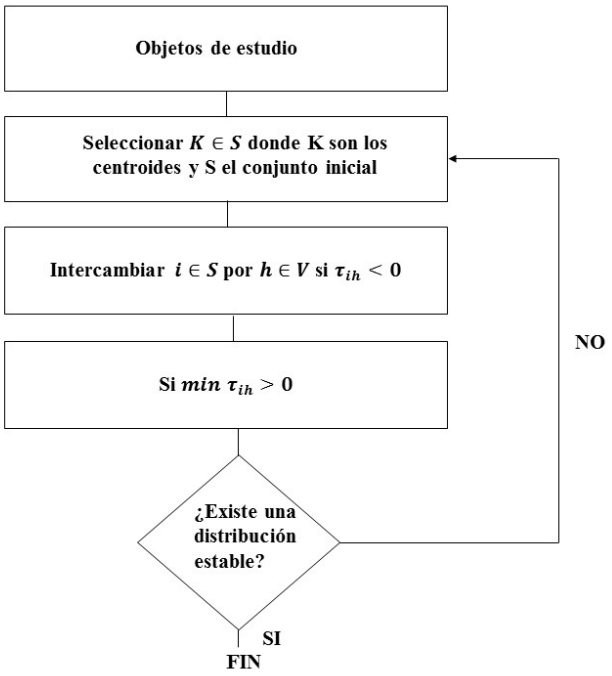

Figura 2.2: Algoritmo *K-medoides*

Usos y Beneficios

Dentro de los Usos y beneficios que tiene este algoritmo se mencionan los siguientes:

1. Se utiliza el algoritmo *K-medoides* sobre todo para reducir los inconvenientes del algoritmo $K - means$, es decir, $K - medoides$ es un método más robusto ante el ruido (datos atípicos) porque minimiza una suma de disimilaridades (entre pares de puntos) en vez de una suma de distancias euclidianas cuadradas. Por lo tanto este método se utiliza sobre todo en matrices de información que tengan muchos datos atípicos que tengan una alta probabilidad de generar ruido en los resultados de los análisis.

2. Funciona bien con conjunto de datos pequeños. Esto es importante en bases de datos clínicos donde no se tienen muchos individuos de estudio.

3. Tiene la capacidad de usar valores no numéricos.

Restricciones

1. Encontrar el valor K es una tarea difícil.

2. No es efectivo cuando se usa con un clúster global, esto debido a que el algoritmo detecta datos atípicos que generan ruido.

3. Si se han seleccionado diferentes particiones iniciales, puede variar el resultado para los clústers.

4. El algoritmo no maneja el clúster de diferente tamaño y densidad.

5. No escala bien debido a su complejidad computacional.

Mejoras

Después de que Kaufman y Rousseeuw propusieran el método K-medoides en 1987, investigadores han generado mejoras para disminuir las restricciones del mismo. Entre esos están:

1. (Rangel, Hendrix, Agrawal, Liao, y Choudhary, 2016), generaron un algoritmo rápido para estimar medoides en grandes conjuntos de datos llamado AGORAS ya que debido a su complejidad computacional asociada, K-medoides, no se aplica a grandes volúmenes de datos. AGORAS es un algoritmo heurístico novedoso para el problema de *K-medoides* donde la complejidad algorítmica es controlada por k, el número de grupos, en lugar de n, el número de puntos de datos.

2. (Yu, Liu, Guo, y Liu, 2018) crearon un algoritmo *K-medoides* mejorado basado en el aumento y la optimización de medoides donde proponen un algoritmo de agrupamiento que preserva la eficiencia computacional y la simplicidad del algoritmo de *K-medoides* simple y rápido al mismo tiempo que mejora su rendimiento de agrupamiento. El algoritmo propuesto requiere la determinación de los subconjuntos de medoides candidatos y el cálculo de la matriz de distancia, luego usar ambos para aumentar incrementalmente el número de medoides agrupados y nuevos de 2 a k, así como seleccionar dos medoides iniciales.

2.1.3. CLARA

$CLARA$ (*Clustering Large Applications*), es el tercer método de segmentación que se estudiará dentro del grupo de los métodos de partición en este trabajo de grado, el cual se encarga de encontrar medoides para todo el conjunto de datos. $CLARA$ considera una pequeña muestra de datos con tamaño fijo (tamaño de muestra) y aplica el algoritmo $K - medoides$ para generar un conjunto óptimo de medoides para la muestra. La calidad de los medoides resultantes se mide por la disimilaridad promedio entre cada objeto en todo el conjunto de datos y el medoide de su grupo, definido como la función de coste.

Los resultados finales de la agrupación corresponden al conjunto de medoides con el coste mínimo.

Definición

Este método combina la idea de K-medoides con el remuestreo para que pueda aplicarse a grandes volúmenes de datos. En lugar de intentar encontrar los medoides empleando todos los datos a la vez, CLARA selecciona una muestra aleatoria de un tamaño determinado y le aplica el algoritmo K-medoides para encontrar las agrupaciones óptimas acorde a esa muestra. Al utilizar esos medoides, se agrupan las observaciones de todo el set de datos. La calidad de los medoides resultantes se cuantifica con la suma total de las distancias entre cada observación

del conjunto de datos y su correspondiente medoide (suma total de distancias intra-clusters).

CLARA repite este proceso un número predeterminado de veces con el objetivo de reducir el sesgo del muestreo. Por último, se seleccionan como clústeres finales los obtenidos con aquellos medoides que han conseguido la menor suma total de distancias. A continuación, se describen los pasos del algoritmo *CLARA*.

Algoritmo

1. Se divide aleatoriamente el conjunto de datos en n partes de igual tamaño, donde n es un valor que determina el analista.
2. Para cada una de las n partes:
   a) Aplicar el algoritmo K-medoides e identificar cuáles son los *K-medoides*.
   b) Al utilizar los medoides del paso anterior, agrupar todas las observaciones del set de datos.
   c) Calcular la suma total de las distancias entre cada observación del conjunto de datos y su correspondiente medoide (suma total de distancias intra-clusters).
3. Seleccionar como segmentación final aquel que ha conseguido la menor suma total de distancias intra-clusters en el paso 2.c.

Usos y beneficios

- Se puede aplicar a grandes volúmenes de datos.
- Utiliza los procesos de muestreo y agrupamiento a un número predeterminado de veces para minimizar el sesgo de muestreo.

Restricciones

- En caso de que la muestra del centroide no se encuentre dentro de los mejores *k-centroides*, el algoritmo no puede encontrar el mejor agrupamiento, por lo que pierde eficiencia.
- Si el tamaño de la muestra no es lo suficientemente grande, la efectividad del algoritmo es baja, y por otro lado si el tamaño de la muestra es demasiado grande el rendimiento del algoritmo no arroja buenos resultados.

Mejoras

- (Ng y Han, 2002) compartieron su investigación sobre el algoritmo CLARANS (*Clustering Large Applications based on Randomized Search*) donde combina las técnicas de muestreo con $K - medoides$. El proceso de agrupamiento se puede presentar como una búsqueda en un gráfico donde cada nodo es una solución potencial, es decir, un conjunto de *K-medoides*. El agrupamiento obtenido después de reemplazar un medoide se denomina vecino del agrupamiento actual. CLARANS selecciona un nodo y lo compara con

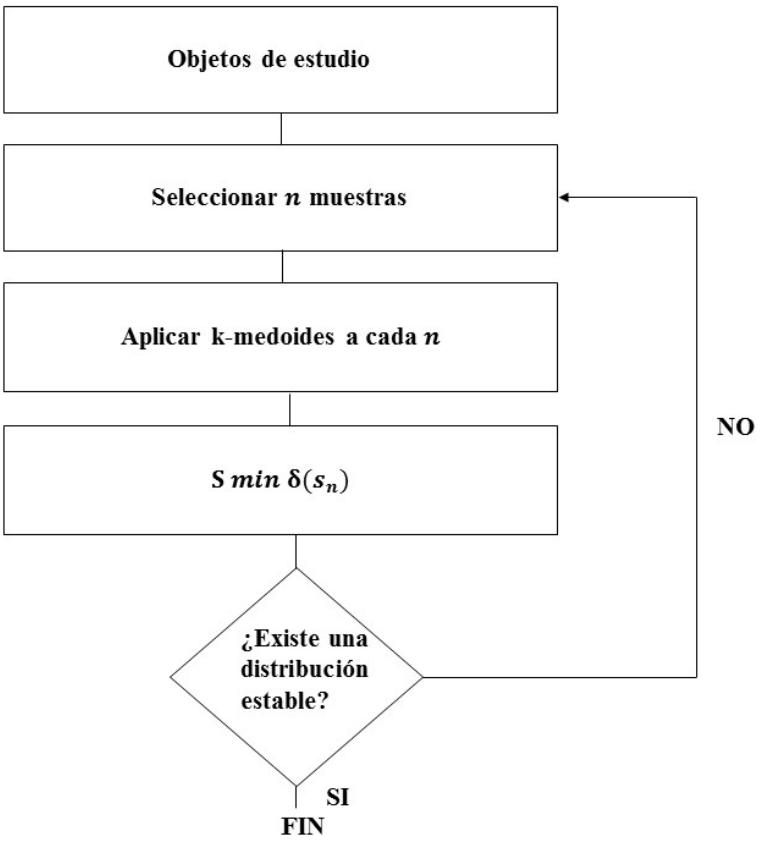

Figura 2.3: Algoritmo *Clara*

un número definido por el usuario de sus vecinos en busca de un mínimo local. Si se encuentra un mejor vecino (es decir, que tiene un error de cuadrado inferior), CLARANS se mueve al nodo del vecino y el proceso comienza de nuevo; de lo contrario, la agrupación actual es un óptimo local. Si se encuentra el óptimo local, CLARANS comienza con un nuevo nodo seleccionado al azar en busca de un nuevo óptimo local.

2.2. Métodos de Densidad

En la agrupación basada en la densidad hay una partición de dos regiones, es decir, región de baja densidad a región de alta densidad.

Un grupo se define como un componente denso conectado que crece en cualquier dirección a la que conduce una densidad. Esta es la razón por la cual los algoritmos basados en la densidad son capaces de descubrir grupos de formas arbitrarias y proporciona protección natural a los valores atípicos. Básicamente, la agrupación basada en densidad se divide en dos categorías, a saber, conectividad basada en densidad y función de densidad.

La densidad y la conectividad son dos conceptos principales que se encuentran bajo que, ambos se miden en términos de distribución local de los vecinos más cercanos. Ejemplos de

algoritmos de conectividad basados en densidad son los algoritmos DBSCAN, GDBSCAN, OPTICS y DBCLASD entre otros y la función de densidad incluye el algoritmo DENCLUE.

DBSCAN, DENCLUE son los cuales estaremos estudiando en esta sección.

2.2.1. DBSCAN

El agrupamiento espacial basado en densidad de aplicaciones con ruido o "Density-based spatial clustering of applications with nois" (DBSCAN) es un algoritmo de agrupamiento de datos (data clustering) (Ester, Kriegel, Sander, Xu, y cols., 1996) siendo uno de los algoritmos de agrupamiento más usados y citados en la literatura científica.

DBSCAN es un algoritmo de agrupamiento basado en densidad (density-based clustering) porque encuentra un número de grupos (clusters) comenzando por una estimación de la distribución de densidad de los nodos correspondientes.

DBSCAN utiliza el concepto de accesibilidad de densidad y conectividad de densidad.

Definición

- **Accesibilidad de densidad**: Se dice que un punto p es de densidad accesible desde un punto q si el punto p está dentro de la distancia ϵ del punto q y q tiene suficiente número de puntos en sus vecinos que están dentro de la distancia ϵ.

- **Conectividad de densidad**: Se dice que un punto p y q están conectados a la densidad si existe un punto r que tiene un número suficiente de puntos en sus vecinos y los puntos p y q están dentro de la distancia ϵ. Este es el proceso de encadenamiento, entonces, si q es vecino de r, r es vecino de s, s es vecino de t, que a su vez es vecino de p y esto implica que q es vecino de p.

Algoritmo

Sea $X = \{x_1, x_2, x_3, ..., x_n\}$ el conjunto de puntos de datos. DBSCAN requiere dos parámetros, ϵ y el número mínimo de puntos necesarios para formar un clúster:

1. Se debe comenzar con un punto de partida arbitrario.

2. Extraer la vecindad de este punto usando ϵ (Todos los puntos que están dentro de la distancia ϵ son vecinos).

3. Si hay suficientes puntos alrededor de este punto, se inicia el proceso de agrupación y el punto se marca como visitado; de lo contrario, este punto se etiqueta como ruido (más tarde, este punto puede convertirse en parte del grupo).

4. Si se encuentra que un punto es parte del grupo, entonces su ϵ -vecindad también es parte del grupo y el procedimiento anterior del paso 2 se repite para todos los puntos de ϵ -vecindario. Esto se repite hasta que se determinen todos los puntos en el grupo.

5. Se recupera y procesa un nuevo punto no visitado, lo que conduce al descubrimiento de otro grupo o ruido.

6. Este proceso continúa hasta que todos los puntos estén marcados como visitados.

Usos y Beneficios

Dentro de las ventajas y Aplicaciones del algoritmo tenemos los siguientes:

1. DBSCAN no es necesario la especificación del número de clústers como si lo requiere $k - means$

2. DBSCAN puede encontrar clústers con formas geométricas arbitrarias. Puede incluso hallar un cluster completamente rodeado (pero no conectado) de otro clúster distinto.

3. DBSCAN tiene noción del ruido y es robusto detectando outliers.

4. DBSCAN requiere solo de dos parámetros y no es susceptible al orden en que se encuentren los puntos dentro de la base de datos.

Restricciones

1. DBSCAN no es enteramente determinista: los puntos borde que son alcanzables desde más de un clúster pueden etiquetarse en cualquiera de estos. Afortunadamente, esta situación no es usual, y tiene un impacto pequeño sobre el clúster. En los puntos núcleo y ruidosos, DBSCAN es determinista. DBSCAN*4 es una variación que trata los puntos borde como ruido, y así logra un resultado completamente determinista, así como una interpretación estadística de las componentes densamente conectadas más consistente.

2. La calidad de DBSCAN depende de la noción de distancia (distance measure) usada en la función regionQuery(P, ϵ). La distancia más usada es la distancia euclidiana. Especialmente para los datos de alta dimensión, por lo que es difícil encontrar un valor adecuado para ϵ. Este efecto, sin embargo, también está presente en cualquier otro algoritmo basado en la distancia euclidiana.

3. DBSCAN no puede agrupar conjuntos de datos bien con grandes diferencias en las densidades, ya que la combinación ϵ -puntos mínimos no se puede escoger adecuadamente para todos los grupos.

Mejoras

1. OPTICS (Ankerst, Breunig, Kriegel, y Sander, 1999) puede verse como una generalización de DBSCAN para múltiples rangos, al reemplazar el parámetro ϵ por el radio máximo de búsqueda. En 2014, el algoritmo fue merecedor del premio a la prueba del tiempo (un reconocimiento dado a algoritmos que han recibido una sustancial atención en la teoría y la práctica) en la conferencia líder de la minería de datos, KDD.

2. HDBSCAN (*Hierarchical Density-Based Spatial Clustering of Applications with Noise*) (McInnes, Healy, y Astels, 2017) realiza DBSCAN sobre diferentes valores de épsilon e integra el resultado para encontrar un agrupamiento que brinde la mejor estabilidad sobre épsilon. Esto permite a HDBSCAN encontrar grupos de densidades variables (a diferencia de DBSCAN) y ser más robusto para la selección de parámetros. La biblioteca también incluye soporte para agrupación robusta de enlace único (Chaudhuri et al. 2014), (Chaudhuri y Dasgupta 2010), detección de valores atípicos GLOSH (Campello et al. 2015) y herramientas para visualizar y explorar estructuras de agrupaciones. Por último, también está disponible el soporte para predicción y agrupamiento suave.

3. Una implementación de DBSCAN así como GDBSCAN y otras variantes están disponibles en ELKI framework. Esta aplicación puede utilizar diversas estructuras de índices de ejecución sub-cuadrática y apoya diversas funciones de distancia y tipos de datos arbitrarios, pero puede ser superado por el bajo nivel optimizado (y especializado) implementándose en pequeños conjuntos de datos.

2.2.2. DENCLUE

El algoritmo "Density Based Clustering" (Denclue) (Hinneburg, Keim, y cols., 1998), utiliza el concepto de influencia y de densidad. En esta influencia de cada punto de datos, se puede modelar formalmente usando una función matemática llamada función de influencia. La función de influencia describe el impacto del punto de datos dentro de su vecindad. Después de eso, calculamos la función de densidad que es la suma de las influencias de todos los puntos de datos.

Definición

De acuerdo con DENCLUE se definen dos tipos de Clústers, es decir, grupos definidos en el centro y en centros múltiples:

1. *El grupo definido en el centro, un atractor de densidad* $x*(f_B^D(X^*) > \epsilon)$ *es el subconjunto de la base de datos que es densidad atraída por* x^*.

2. *El clúster definido multicéntrico, consiste en un conjunto de clústeres definidos en el centro que están unidos por una ruta con significado y* ϵ *es el umbral de ruido.*

La función de influencia $f_B^y : F^d \to R_0^+$ de un objeto de datos $y \in F^d$ es una función que se define en términos de una función de influencia básica f_B

$$f_B^y(x) = f_B(x, y) \tag{2.4}$$

La función de densidad se define como la suma de las funciones de influencia de todos los puntos de datos. Dados N objetos de datos descritos por un conjunto de vectores de características $D = \{x_1, x_2, x_3, ..., x_n\} \in F^d$ la función de densidad es definida como

$$f_B^D = \sum_{i=1}^{N} f_B^{x_i}(x) \tag{2.5}$$

Algoritmo

El algoritmo DENCLUE funciona en dos pasos.

1. El primer paso es un paso previo a la agrupación, en el que se construye un mapa de la parte relevante del espacio de datos. El mapa se utiliza para acelerar el cálculo de la función de densidad que requiere acceder de manera eficiente a las partes vecinas del espacio de datos.

2. El segundo paso es el paso de agrupamiento real, en el que el algoritmo identifica los atractores de densidad y los puntos atraídos por la densidad correspondiente.

En el siguiente diagrama especificamos los pasos:

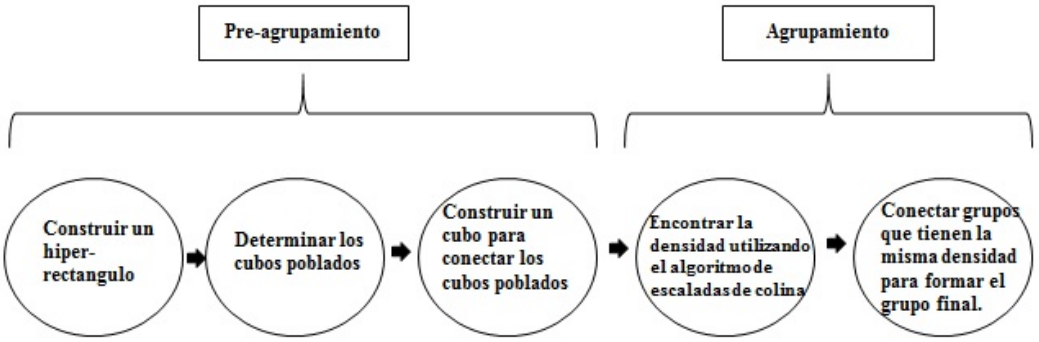

Figura 2.4: Algoritmo Denclue

Usos y Beneficios

1. Logra buenos agrupamientos en matrices de información con puntos ruidosos.

2. Es significativamente más rápido que otros algoritmos de agrupamiento.

Restricciones

1. Problema de la selección de sus parámetros σ y ξ, con σ determina la influencia de un punto en su vecindad y ξ es el umbral de densidad.

Mejoras

El algoritmo DENCLUE-IM (Rehioui, Idrissi, Abourezq, y Zegrari, 2016) es una mejora de DENCLUE. Esta mejora consiste en modificar el paso basado en el algoritmo de Hill Climbing. Este paso considerado crucial en el algoritmo DENCLUE se basa en cálculos de gradiente. Estos cálculos se realizan en cada punto para encontrar su atractor de densidad. Hacer cálculos para cada punto no óptimo para lograr resultados en un tiempo razonable, especialmente cuando se trata de operar en grandes matrices de información. Este enfoque permite encontrar un elemento equivalente al atractor de densidad, que representará todos los puntos contenidos en un hipercubo, en lugar de los cálculos realizados para cada punto en el conjunto de datos. Este

representante del hipercubo denotado x_{Hcube}, se considerará como el punto que tiene la mayor densidad en este hipercubo como se muestra:

$$\forall\, x \in C_p,\ f_D(x) \leq f_D(x_{Hcube})$$

Donde C_p es un hipercubo poblado dado en el hiper-rectángulo construido. Por lo tanto, cada hipercubo constituye un grupo inicial representado por su x_{Hcube}. Estos clústers x_{Hcube} se fusionarán, si y solo sí, existe un camino entre sus representantes.

2.3. Métodos difusos

Los algoritmos descritos anteriormente resultan en grupos nítidos, lo que significa que un punto de datos pertenece a un grupo o no. Los clústeres no se superponen y este tipo de partición se denomina además agrupamiento nítido. La cuestión del soporte de incertidumbre en la tarea de agrupamiento lleva a la introducción de algoritmos que utilizan conceptos de lógica difusa en su procedimiento.

Estos métodos difusos, surgen con la necesidad de resolver deficiencia del agrupamiento, donde se considera que cada elemento se agrupa de manera equivocada con los elementos de su clúster y por lo tanto no es similar al resto de sus elementos. Tras la introducción de la lógica difusa (Zadeh, 1965) surgió una solución para este problema, se caracteriza la similitud de cada uno de los elementos a cada uno de los clústers. De acuerdo a la función de pertenencia, se toman valores entre cero y uno. Los valores más cercanos a uno indican que existe mayor similitud y para los valores más cercanos a cero indican una menor similitud. Por lo que el problema en los metodos difusos, se reducen a encontrar una caracterización de tipo que sea óptima.

Una partición difusa caracteriza la participación de cada muestra en todos los grupos utilizando funciones de pertenencia que toman valores entre cero y uno. Además, cumplen que para cada muestra la suma de sus participaciones en cada grupo sea uno. De esta forma, es posible traducir el problema del agrupamiento difuso en encontrar una partición difusa óptima. A continuación se encuentra una definición más formal de este concepto.

Sea $X = (x_1, ..., x_n)$ un subconjunto de un espacio euclidiano de dimensión s y sea c un entero positivo mayor que uno. Una partición difusa de X en c grupos es una tupla de c funciones de pertenencia $\mu = (\mu_1, \ldots, \mu_c)$ que cumplen las siguientes condiciones:

1. $0 \leq \mu_i(x) \leq 1, \forall i, i = 1, 2, \ldots, C$
2. $0 < \sum_{j=1}^{n} \mu_i(x_j) < 1, \forall i$
3. $\sum_{i=1}^{c} \mu_i(x_j) = 1, \forall j$

Las particiones difusas se representan como una matriz asociando cada fila a uno de los c grupos y cada columna a uno de los elementos de X, de forma tal que, el valor en la fila i y la

columna j indique la pertenencia del elemento j al grupo i. Más formalmente, el conjunto de las particiones difusas se puede definir como:

$$M_{fc} = \{U \in \Re^{c \times n} | U = [u_{ij}]; u_{ij} \in [0,1] \forall i, j; \sum_{i=1}^{c} u_{ij} = 1 \forall j; \sum_{j=1}^{n} u_{ij} > 0 \forall i\} \tag{2.6}$$

2.3.1. Método Fuzzy C-means

Seún estudio realizado por (Bellido, 2017), en 1973, Bezdek y Dunn presentaron un método de clustering que combinaba los conceptos de los métodos basados en función objetivo con los de la lógica Fuzzy.

Los algoritmos basados en clasificación Fuzzy son ampliamente utilizados en aplicaciones de compresión, organización y clasificación de información.
En la actualidad se viene realizando investigaciones sobres la base del algoritmo FCM, en especial para la aplicación de reconocimiento de patrones para imágenes dentro del ámbito médico, y geográfico, de la misma manera para minería de datos con Fuzzy Miner y en la BIGDATA adaptando el método Fuzzy a diferentes métodos para obtener resultados robustos.

Definición

Sea $X = x_1, \ldots, x_n \subset R^S$ un conjunto de datos (los reales) p-dimensional cada uno, se dice que una partición $P = c_1, c_2, \ldots, c_c$ donde $U = [u_{ik}] \in R^{cn}$ c es una partición suave de X si y solo si se cumple que:

1. $u_{ik} \in [0,1]$; $1 \leq i \leq c$; $1 \leq k \leq n$
2. $\sum_{i=1}^{c} u_{ik} = 1$; $1 \leq k \leq n$
3. $\sum_{k=1}^{n} u_{ik} > 0$; $1 \leq i \leq c$

El conjunto de todas las matrices R^{cn} satisfaciendo (1) es denotado por M_{fcn} una matriz $U \in M_{fcn}$ puede ser usado para describir la estructura de clúster de X interpretando u_{ik} como el grado de pertenencia de x_k para el clúster i, miéntras $u_{ik} = 0,95$ representa una fuerte asociación de x_k para el clúster i un $u_{ik} = 0,01$ representa una muy débil.

- En (1) vemos como el grado de pertenencia de un objeto k a un clúster i debe estar entre 0 y 1.
- En (2) vemos como la suma de los grados de pertenencia de un objeto k a los distintos grupos ha de ser igual a 1, al cumplir esta condición más es llamada una partición suave restringida.
- En (3) vemos como la suma de todos los grados de pertenencia en un clúster tiene que ser mayor a 0 y menor que n es decir no podemos tener grupos vacíos ni un clúster con todos los elementos.

Tomamos a M_{cn} como el subconjunto de M_{fcn} que contiene solo matrices con todos los u_{ik} en $0, 1$, es exactamente el conjunto de no degenerado crip (o convencional) c-particiones de X.

Otra información útil acerca de la subestructura de clúster puede ser idetificada para prototipos (o centros de clúster) $v = v_1, \ldots, v_c{}^T \in \mathrm{R}^{CS}$ donde v_i es prototipo para clase i, $1 \leq i \leq c$, $v \in R^S$. Buenas particiones U de X y representativos (v para i clases) puede ser definido considerando minimización de una de la familia $c - means$ objetivos funcionales $J : (M_{fcm} \times R^{cs}) \longrightarrow R$ definido por:

$$J_m(U, v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m \parallel x_k - y_i \parallel^2 \tag{2.7}$$

Donde m es el factor difuso que indica cuanto queremos que se solapen los grupos y tiene que cumplir que $1 \leq m \leq \infty$ y $\parallel . \parallel$ es alguna norma inducida sobre R^S. Esta aproximación fue primero dada por $m = 2$ en Dun(1973) y entonces generalizado para el rango de valores de m en Bezdek (1973).

Algoritmo

De acuerdo a lo anterior, el procedimiento general de los algoritmos *Fuzzy C-Means*, se pueden formalizar en los siguientes pasos:

1. Se debe fijar los valores de c, m, A y $||k||_A$. Elegir una matriz inicial $U^{(0)} \in M_{fc}$.
2. Se debe calcular los centros de los grupos con la fórmula

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m}; 1 \leq i \leq c \tag{2.8}$$

3. Luego se debe actualizar la matriz de partición difusa $U = [u_{ik}]$ con,

$$u_{ik} = (\sum_{j=1}^{c} (\frac{d_{ik}}{d_{jk}})^{\frac{2}{m-1}})^{-1}; 1 \leq k \leq n; 1 \leq i \leq c \tag{2.9}$$

4. Si se alcanzó el criterio de parada, terminar. En caso contrario, regresar al paso 2.

Algunos de los criterios de parada más utilizados son:

1. Un número máximo de iteraciones
2. Que la variación en la matriz U sea muy pequeña: $||U^{k+1} - U^k|| < \epsilon$.

Usos y beneficios

- Estos algoritmos son útiles en matrices de información donde no se tienen claros los resultados de la agrupación "nítida" ya que hay datos que por sus carácteristicas pueden pertenecer a mas de un cluster.
- Estos algoritmos se han aplicado ampliamente en diferentes áreas como el procesamiento de imágenes, sistemas de ingeniería, estimación de parámetros, entre otras.

Restricciones

- Mal comportamiento de los datos cuando se presenta mucho ruido.
- Hay que especificar el número de clústers.

Mejoras

- Los algoritmos *Possibilistic C-means* (Krishnapuram y Keller, 1996) aparecen con el objetivo de resolver el mal comportamiento de los algoritmos *Fuzzy C-means* al ser utilizados en conjuntos de datos con mucho ruido. Estos algoritmos se caracterizan por interpretar los valores u_{ij} como grados de compatibilidad con los grupos, en lugar de probabilidades. Para ello, se relaja la restricción de las particiones difusas que obliga a que la suma de los grados de pertenencia de un elemento hacia todos los grupos sea uno, exigiendo solamente que al menos uno de los grados de pertenencia sea positivo.

  Por lo tanto, las restricciones en la definición de partición difusa podrían reescribirse como:

  1. $0 \leq \mu_i(x) \leq 1, \forall i = 1, \ldots, c$
  2. $0 < \sum_{j=1}^{n} \mu_i(x_j) < 1, \forall i$
  3. $\max_i \mu_i(x_j) = 1, \forall j$

  Una de las funciones objetivos más utilizadas por estos algoritmos es la siguiente:

$$J_m(U, v, \eta) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m d_{ik}^2 + \sum_{i=1}^{c} \eta_i \sum_{k=1}^{n} (1 - u_{ik})^m \tag{2.10}$$

  La función (2.10) es la misma función objetivo de los algoritmos *Fuzzy C-means* con un término añadido que impide que la partición obtenida sea la solución trivial donde todos los valores de pertenencia sean iguales a cero. El vector $\eta = (\eta_1, \eta_2, \ldots, \eta_c)$ es un vector de valores positivos, donde sus valores η_i denotan la distancia desde el centro del grupo i a la del grado de pertenencia de un elemento y sea 0.5. Estos valores determinan el

tamaño y forma de sus grupo correspondiente y generalmente se calculan utilizando la siguiente fórmula:

$$\eta_i = K \frac{\sum_{j=1}^{n} u_{ij}^m d_{ij}^2}{\sum_{j=1}^{n} u_{ij}^m} \tag{2.11}$$

donde K es normalmente uno.

- Se realizó una adaptación útilizando el método *Fuzzy C-means* donde se incoporó el proceso de selección de atributos. Para ello se introduce la métrica de distancia a utilizar, la forma de replantear el problema a optimizar y su nueva solución. Luego se discute acerca del algoritmo resultante, para finalmente proponer un esquema de regularización de los pesos de los atributos y con ello ajustar la complejidad del modelo resultante.

Capítulo III
INDICES DE VALIDACIÓN DE CLÚSTERES

Como se mencionó en el capítulo anterior, la segmentación de datos, se encarga de hacer un análisis de perfilamiento de los datos, con el objetivo de encontrar patrones y entender el comportamiento de estos.

Uno de los problemas más importantes en el análisis de segmentación, es la evaluación de los resultados de las agrupaciones para encontrar la partición óptima que se ajuste a los datos subyacentes. Este es el tema principal de la validez de cluster. A continuación se estudiarán los conceptos fundamentales de estos métodos de evaluación miéntras se presentan los diversos enfoques de validez de grupos propuestos en la literatura.

En la mayoría de las evaluaciones experimentales de los algoritmos, los conjuntos de datos 2D se utilizan para que el investigador pueda verificar visualmente la validez de los resultados (es decir, que tan bien el algoritmo de agrupación determinó los grupos del conjunto de datos que se adaptan al perfilamiento adecuado). Está claro que la visualización del conjunto de datos es una verificación crucial de los resultados del agrupamiento. En el caso de grandes conjuntos de datos multidimensionales (por ejemplo, más de tres dimensiones) la visualización efectiva del conjunto de datos no es posible. Por lo tanto, la percepción de los clústeres es una tarea difícil de visualizar en dimensiones superiores.

Los diferentes algoritmos de agrupamiento se comportan de una manera diferente dependiendo de:

- Las características del conjunto de datos (geometría y distribución de densidad de los grupos)
- Los valores de los parámetros que se tienen en cuenta en la entrada

Por ejemplo, suponga el conjunto de datos (a) en la figura (3.1):

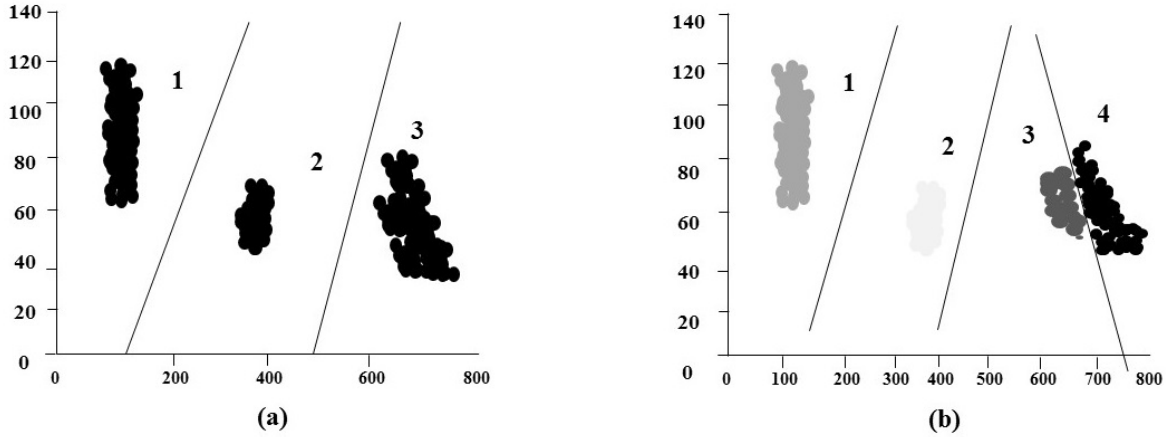

Figura 3.1: Conjunto de datos

en donde la figura 3.1 (a) se tiene Un conjunto de datos que consta de 3 grupos y la figura 3.1 (b) Los resultados de la aplicación de K-medias cuando pedimos cuatro grupos.

Como lo menciona (Maria Halkidi, 2001), se puede descubrir tres grupos en el conjunto de datos dado. Sin embargo, si se considera un algoritmo de agrupación en clústeres (por ejemplo, K-Means) con ciertos valores de parámetros (en el caso de K-means, el número de agrupaciones) para dividir el conjunto de datos en cuatro agrupaciones, el resultado del proceso de agrupación sería el esquema de agrupación presentado en la figura (b). En nuestro ejemplo, el algoritmo de agrupación *K-means* encontró los cuatro mejores clústeres en los que el conjunto de datos podría dividirse. Sin embargo, esta no es la partición óptima para el conjunto de datos considerados. Aquí, se define, el término esquema de segmentación "óptimo", como el resultado de ejecutar un algoritmo de segmentación (es decir, una partición) que se ajuste mejor a las particiones inherentes del conjunto de datos. Es obvio que, el esquema representado en la figura (b) no es el mejor para el conjunto de datos, es decir, el esquema de clusterización presentado no se ajusta bien al conjunto de datos. El agrupamiento óptimo para el conjunto de datos será un esquema con tres agrupaciones.

Como consecuencia, si a los parámetros del algoritmo de agrupación se les asigna un valor incorrecto, el método de agrupación puede dar como resultado un esquema de partición que no es óptimo para el conjunto de datos específico que conduce a decisiones erróneas.

Los problemas de decidir el número de agrupaciones que se ajustan mejor a un conjunto de datos así como la evaluación de los resultados de agrupación han sido objeto de varios esfuerzos de investigación (Dave, 1996) (Gath y Geva, 1989) (Rezaee, Lelieveldt, y Reiber, 1998) (U. M. Fayyad, Piatetsky-Shapiro, Smyth, y cols., 1996) (Theodoridis y Koutroumbas, 1999) (Xie, 1991).

A continuación, se discuten los conceptos fundamentales de la validez de agrupación y se muestran los criterios más importantes.

3.1. Conceptos Fundamentales

Hay dos criterios propuestos para la evaluación de la agrupación y la selección de un esquema de agrupación óptimo (Berry y Linoff, 1997):

1. **Cohesión**. Los miembros de cada agrupación deben estar lo más cerca posible entre sí. Una medida común de la Cohesión es la varianza, que debe minimizarse.

2. **Separación**. Los clústeres deben estar ampliamente separados entre ellos. Existen varios enfoques para medir esta distancia entre clústers: distancia entre el miembro más cercano, distancia entre los miembros más distantes o la distancia entre los centroides:

    - *Enlace único*: Mide la distancia entre los miembros más cercanos de los grupos.
    - *Enlace completo*: Mide la distancia entre los miembros más distantes.
    - *Comparación de centroides*: Mide la distancia entre los centros de los grupos.

Los dos primeros puntos se basan en pruebas estadísticas y su principal inconveniente es su alto costo computacional. Además, los índices relacionados con estos enfoques apuntan a medir el grado en que un conjunto de datos confirma un esquema especificado a-priori. Por otro lado, el tercer enfoque apunta a encontrar el mejor esquema de agrupamiento para que un algoritmo de segmentación pueda definirse bajo ciertos supuestos y parámetros.

3.2. Generalización

El procedimiento para evaluar los resultados de un algoritmo de agrupamiento se conoce bajo el término de "validez" de clúster. En términos generales, hay tres enfoques para investigar la validez de un clúster (Theodoridis y Koutroumbas, 1999) :

- *Criterios Externos*. Esto implica que se evalúan los resultados de un algoritmo de agrupamiento con base en una estructura pre-especificada, que se impone a un conjunto de datos y refleja la intuición sobre la estructura de agrupamiento de datos del conjunto.

- *Criterios Relativos*. Aquí la idea básica, es la evaluación de una estructura de agrupación de clúster comparándola con otros esquemas de agrupación de clúster, que resultan del mismo algoritmo pero con diferentes valores de parámetros.

- *Criterios Internos*. Podemos evaluar los resultados de un algoritmo de agrupación en términos de cantidades que involucran los vectores del conjunto de datos (por ejemplo, matriz de proximidad).

Se presentará a continuación un análisis en cada enfoque, pero se discutirán a profundidad algunos métodos de criterios internos (índices de validación internos) debido a que son los métodos más utilizados para validar algoritmos de segmentación para datos no supervisados. Sin embargo, debemos mencionar que estos métodos dan una indicación de la calidad de la partición resultante y, por lo tanto, solo pueden considerarse como una herramienta a disposición de los expertos para evaluar los resultados de la segmentación.

3.3. Indices de Validación Externos

En este enfoque, la idea básica es probar si los puntos del conjunto de datos están estructurados aleatoriamente o no, basados en pruebas estadísticas. Este análisis se basa en la hipótesis nula:

- H_o *Sea X, un conjunto de datos con estructura aleatoria*

Dichas pruebas estadísticas, tienen un trabajo cumputacionalmente complejo, por lo tanto, las técnicas de Monte Carlo se utilizan para solucionar dichos problemas para validar los algoritos de clusterización. (Theodoridis y Koutroumbas, 1999).

Las técnicas de Monte Carlo se utilizan principalmente para calcular la función de densidad de probabilidad de los índices estadísticos definidos. Primero, se debe generar una gran cantidad de conjuntos de datos sintéticos (artificiales) aleatorios. Para cada uno de estos conjuntos de datos sintéticos, llamados X_i, se debe calcular el valor del índice definido, denotado q_i. Luego, con base en los valores respectivos de q_i para cada uno de los conjuntos de datos X_i, se crea un diagrama de dispersión. Este diagrama de dispersión es una aproximación de la función de densidad de probabilidad del índice. En la figura (3.2) se denotan los tres casos posibles de la forma de la función de densidad de probabilidad de un índice q. Existen tres formas posibles diferentes según el intervalo de crítico $\overline{D_\rho}$, correspondiente al nivel significativo ρ (constante estadístico). Como se puede observar, la función de densidad de probabilidad de un índice estadístico q, bajo H_o, tiene un máximo único y la región $\overline{D_\rho}$ es una semirecta o una unión de dos semirectas.

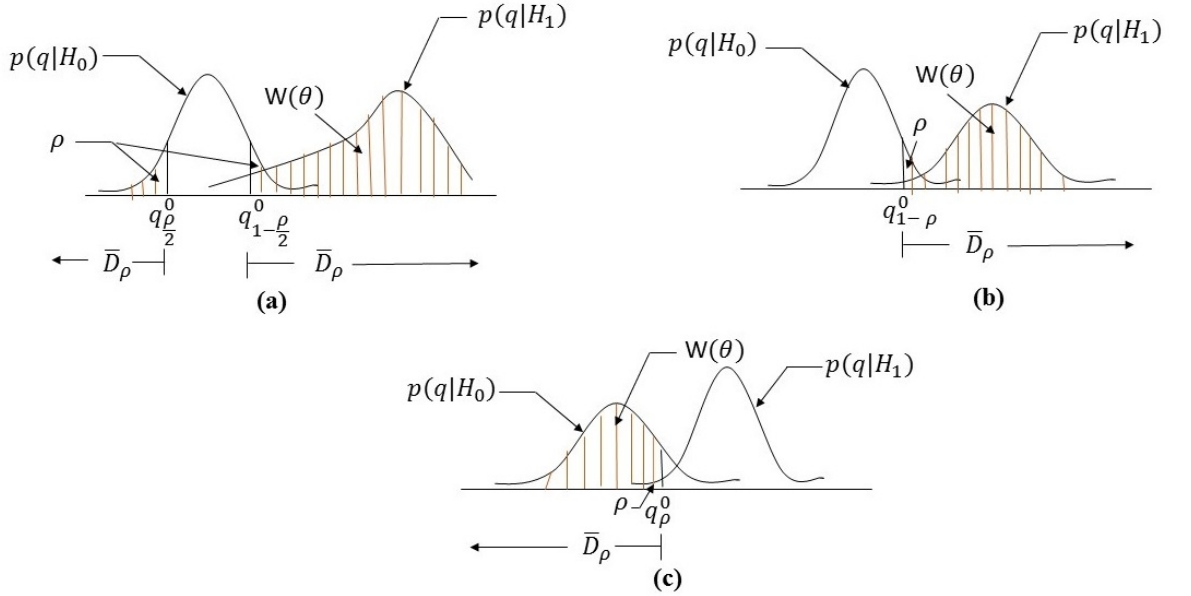

Figura 3.2: Indices Externos

Donde donde q_p^0 es la proporción ρ de q bajo la hipótesis H_o (Theodoridis y Koutroumbas, 1999) y

(a) es el índice de dos colas

(b) el índice de cola derecha

(c) es el índice de cola izquierda.

Si la forma de la función de densidad de probabilidad se asemeja a la de cola derecha como muestra la figura 3.2(b) y que se ha generado el diagrama de dispersión utilizando valores r del índice q, llamado q_i, para aceptar o rechazar la Hipótesis nula H_o, se examinan las siguientes condiciones (Theodoridis y Koutroumbas, 1999):

- Se rechaza H_o, si el valor de q para el conjunto de datos es mayor que $(1- >)\cdot r$ de los valores de q_i, de los respectivos conjuntos de datos sintéticos X_i.
- Suponiendo que la forma es de cola izquierda de la figura 3.2(c), se rechaza H_o si el valor de q para el conjunto de datos es menor que $>\cdot\ r$ de los valores de q_i.
- Suponiendo que la forma tiene dos colas de la figura 3.2(a), se acepta H_o si q es mayor que $(> /2)\cdot r$, número de valores de q_i y menor que $(1- > /2)\cdot r$ de valores de q_i.

3.4. Indices de Validación Relativos

La base de los métodos de validación Externo e Interno es la prueba estadística. Por lo tanto, el principal inconveniente de las técnicas basadas en estos criterios, es su alta demanda computacional. Los índices de validación relativos se basan en criterios relativos y no implican pruebas estadísticas.

La idea fundamental de este enfoque es elegir el mejor grupo de segmentación de un conjunto de esquemas definidos de acuerdo con un criterio pre-especificado. De manera concreta, el problema se puede expresar de la siguiente forma:

"Sea P_{alg}, el conjunto de parámetros asociados con un algoritmo de segmentación específico (por ejemplo, la cantidad de clústeres nc). Entre los esquemas de agrupamiento C_i con $i = 1, ..., nc$, definido por un algoritmo específico, para diferentes valores de los parámetros en P_{alg}, se elije el que mejor se ajuste al conjunto de datos".

Entonces, se puede considerar los siguientes casos del problema:

I. P_{alg} no contiene el número de clústers, nc, como parámetro. En este caso, la elección de los valores de los parámetros óptimos se describe a continuación:

   a) Se ejecuta el algoritmo para una amplia gama de valores de sus parámetros y se debe elegir el rango más grande para el cual nc permanece constante (usualmente $nc << N$ (número de tuplas) lo que quiere decir nc multiplicado por 2, N veces).

   b) Luego, se debe seleccionar como valores apropiados de los parámetros de P_{alg} los valores que corresponden a la mitad de este rango. Además, este procedimiento identifica el número de grupos que subyacen en nuestro conjunto de datos.

II. P_{alg} contiene nc como parámetro. El procedimiento, para identificar el mejor esquema de segmentación, se basa en un índice de validez. Seleccionando un índice de rendimiento adecuado q y que procede con los siguientes pasos:

   a) Se ejecuta el algoritmo de agrupamiento para todos los valores de nc entre un nc_{min} mínimo y un nc_{max} máximo. Los valores mínimo y máximo han sido definidos a-priori por el usuario.

   b) Para cada uno de los valores de nc, se ejecuta el algoritmo $r\text{-}veces$, utilizando diferentes conjuntos de valores para los otros parámetros del algoritmo (por ejemplo, diferentes condiciones iniciales).

   c) Posterior a lo anterior se deben trazar los mejores valores del índice q obtenidos por cada nc como función de nc.

Con base en lo anteriormente mencionado, se puede identificar el mejor método de segmentación. Se debe enfatizar que existen dos enfoques para definir el mejor agrupamiento en función del comportamiento de q con respecto a nc. Por lo tanto, si el índice de validez no muestra una tendencia creciente o decreciente a medida que nc aumenta, se busca el máximo (mínimo) de la gráfica.

Por otro lado, para los índices que aumentan (o disminuyen) a medida que crecen el número de clusteres, se busca los valores de nc en los que se produce un cambio local significativo en el valor del índice.

3.5. Indices de Validación Internos

Al usar este enfoque de validez de clúster, el objetivo es evaluar el resultado del clúster de un algoritmo usando solo cantidades y características inherentes al conjunto de datos. Hay dos casos en los que se aplican criterios internos de validez de clúster dependiendo de la estructura del agrupamiento (Maria Halkidi, 2001):

a) *Validez de Jerarquía de esquemas de clúster*. Una matriz llamada *matriz Cofenética* P_c puede representar el diagrama de jerarquía que produce un algoritmo jerárquico. El elemento $P_c(i,j)$ de la matriz $Cofentica$ representa el nivel de proximidad en el que los dos vectores x_i y x_j se encuentran en el mismo grupo por primera vez. Se puede definir un índice estadístico para medir el grado de similitud entre las matrices P_c y P (matriz de proximidad). Este índice se llama *coeficiente de correlación Cofenética* y se define como:

$$CPCC = \frac{(\frac{1}{M})\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}c_{ij} - \mu_p\mu_c}{[\sqrt{[(\frac{1}{M})\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}^2 - \mu_P^2][(\frac{1}{M})\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c_{ij}^2 - \mu_c^2]]}, \tag{3.1}$$

donde,

$$-1 \leq CPCC \leq 1,$$

y $M = N * (N-1)/2$ y N es el número de puntos en un conjunto de datos. Además, μ_p y μ_c son los valores medios de las matrices P y P_c respectivamente, y están dados por las siguientes ecuaciones:

$$\mu_p = (\frac{1}{M}) \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} P(i,j) \tag{3.2}$$

$$\mu_c = (\frac{1}{M}) \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} P_c(i,j) \tag{3.3}$$

Además, d_{ij}, c_{ij} son los elementos (i,j) de las matrices P y P_c, respectivamente. Un valor del índice cercano a 0 es un indicador de una similitud significativa entre las dos matrices. El procedimiento de las técnicas de Monde Carlo también se utiliza en este caso de validación.

b) *Validando un Esquema de agrupamiento único*. El objetivo aquí es encontrar el grado de acuerdo a un determinado esquema de agrupación C, que consta de nc clústers y la matriz de proximidad P. El índice definido para este enfoque es el estadístico de Hubert τ (o estadístico normalizado τ). Se utiliza una matriz adicional para el cálculo del índice, que es $Y(i,j) = \{1,\ si\ x_i\ y\ x_j\ pertenecen\ adiferentes\ grupos,\ y\ 0,\ de\ lo\ contrario\}$, $\forall$ $i,j = 1,...,N$.

La aplicación de las técnicas de Monde Carlo, también aquí, es la forma de probar la hipótesis aleatoria en un conjunto de datos determinado.

Ahora teniendo claro los diferentes enfoques de la evaluación de los clústers se presentan con mayor detenimiento algunos índices de validación internos, por dos razones:

1. Son los más utilizados en el campo profesional.
2. Se estarán usando en el próximo capítulo.

3.5.1. Índice Silueta

El coeficiente de Silueta (Petrovic, 2006) es una métrica para evaluar la calidad del agrupamiento obtenido con algoritmos de segmentación. El objetivo de Silueta es identificar cuál es el número óptimo de agrupamientos.

En los algoritmos de aprendizaje no supervisado, la cantidad de grupos puede ser un parámetro de entrada del algoritmo o puede ser determinado automáticamente por el algoritmo. En el primer caso, como ocurre con el algoritmo de *K-means*, la determinación del número óptimo de clústeres tiene que ser realizado mediante alguna medida externa al algoritmo.

Definición

El coeficiente de Silueta, es un indicador que identifica el número mínimo de grupos óptimos que se necesitan para segmentar un conjunto de datos utilizado.

El coeficiente de Silueta para una observación i se denota como $s(i)$ y se define como (Banchero, 2015):

$$s(i) = \frac{b - a}{max\{(a, b\}} \tag{3.4}$$

donde:

- a es el promedio de las distancias de la observación i con las demás observaciones del clúster.
- b es la distancia mínima a otro clúster que no es el mismo en el que está la observación i. Ese clúster es la segunda mejor opción para i y se lo denomina vecindad de i.

El valor de $s(i)$ puede ser obtenido combinando los valores de a y b como se muestra a continuación:

$$s(i) = \begin{cases} 1 - \frac{a}{b} & \text{si} a < b \\ 0 & \text{si} a = b \\ \frac{b}{a} - 1 & \text{si} a > b \end{cases}$$

El coeficiente de Silueta es un valor que varía entre -1 y 1.

Se analizan las posibles soluciones, para que el coeficiente de Silueta sea cercano a 1, el valor de b tiene que ser mayor al de a. Esto significa que la distancia de la observación i a los clusters vecinos es suficientemente grande para que su pertenencia al clúster actual sea la correcta. Es decir, no es similar a sus vecinos.

Un valor de $s(i)$ que sea cercano a cero indica que la observación i está en la frontera de dos clústers.

Y si el valor de $s(i)$ es negativo, entonces la observación i debería ser asignada al cluster más cercano.

Resumiendo:

- $s(i) \approx 1$, la observación i está bien asignada a su clúster.
- $s(i) \approx 0$, la observación i está entre dos clúster.
- $s(i) \approx -1$, la observación i está mal asignada a su clúster.

Se puede calcular el coeficiente de Silueta como el promedio de todos los $s(i)$ para todas las observaciones del conjunto de datos:

$$S = \frac{1}{n} \sum s(i) \tag{3.5}$$

Este coeficiente suele generalmente ser más alto para grupos convexos, bien separados y con una densidad alta.

Es posible realizar una interpretación visual del cálculo de Silueta como se muestra en la siguiente gráfica

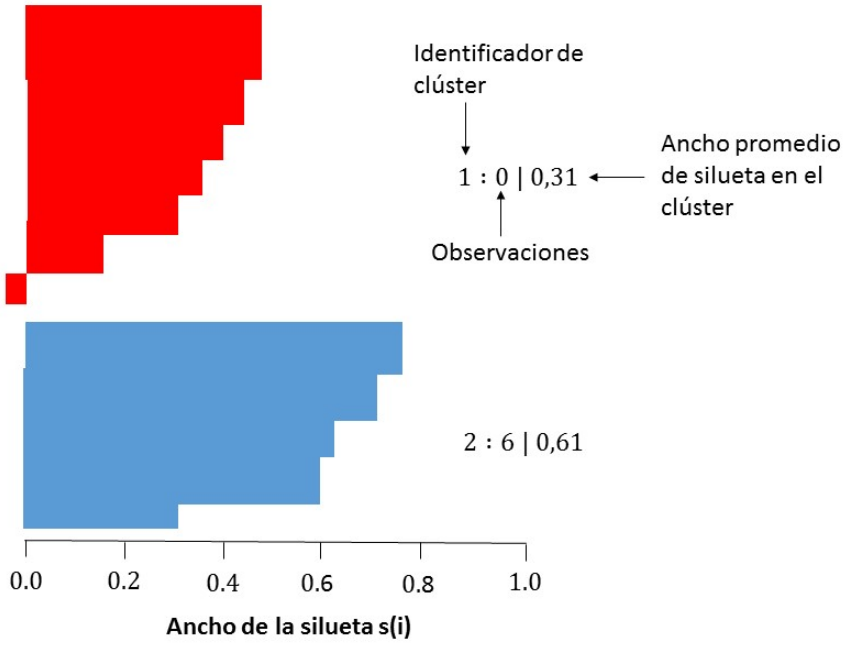

Figura 3.3: Indice Silueta

A partir del análisis visual de este gráfico es posible determinar cuál es el número correcto de grupos en el conjunto de datos analizado. El gráfico de silueta puede ser analizado como un gráfico de barras horizontales donde cada una de las barras representa una observación i para la cual se calculó $s(i)$ que se muestra en el eje horizontal.

En la figura (3.3) del ejemplo realizado para un $K = 2$ se muestra el identificador de clúster, la cantidad de observaciones que lo componen y el ancho de silueta promedio dentro del clúster.

Se observa que para el clúster superior (en rojo) una de las instancias tiene un valor de s(i) menor a cero, es decir, está mal asignado a ese cluster. Esto permite intuir que $K = 2$ no es el

número correcto de clusters para este conjunto de datos, y por lo tanto se tiene que analizar con otros K diferentes.

3.5.2. Indice DB (Davies-Bouldin)

El índice (Davies y Bouldin, 1979) evalúa los algoritmos de agrupamiento y garantiza que el mínimo del valor del índice se determina como el número óptimo de clústers (Qinpei Zhao, 2009).

Definición

Éste índice está definido como:

$$S_i = \frac{1}{k} \sum_{i=1, i \neq j}^{k} \left(\frac{\sigma_i + \sigma_j}{d(c_i, cj} \right) \tag{3.6}$$

Donde k es el número de clústeres, σi es la distancia promedio entre cada punto en el clúster i y el centroide del clúster, σ_j es la distancia promedio entre cada punto del clúster j y el centroide del clúster, y $d(c_i, cj)$ es la distancia entre los centroides de los 2 clústeres.

Valores pequeños para el índice DB indica clústeres compactos, y cuyos centros estas bien separados los unos de los otros.

3.5.3. Índice de Dunn

El índice de Dunn (Dunn, 1974) es la relación entre la distancia más pequeña entre las observaciones que no están en el mismo grupo y la distancia más grande dentro del grupo. El índice de Dunn tiene un valor entre cero e infinito y debe maximizarse.

Definición

El índice Dunn es otra medida de validación interna que se obtiene de la siguiente forma:

sea d_{min} la distancia mínima entre puntos de diferentes conglomerados y d_{max} la mayor distancia dentro del clúster. La distancia entre los clusters C_k y $C_{k'}$ se mide por la distancia entre sus puntos más cercanos, (Bernard, 2017)

$$d_{kk'} = \min_{i \in I_k j \in Ik'} \parallel M_i^k - M_j^{k'} \parallel \tag{3.7}$$

y d_{min} es la más pequeña de estas distancias $d_{kk'}$

$$d_{min} = \min_{k \neq k'} d_{kk'} \tag{3.8}$$

para cada cluster C_K tomando como D_k la mayor distancia que separa dos puntos distintos en el grupo (a veces llamado el diámetro del grupo).

$$D_k = \underset{i,j \in I_k i \neq j}{\text{máx}} \parallel M_i^k - M_j^k \parallel \tag{3.9}$$

entonces d_{max} es la mayor de estas distancias D_k

$$d_{max} = \underset{1 \leq k \leq K}{\text{máx}} D_k \tag{3.10}$$

De acuerdo a lo anterior, el índice de dunn se define como el cociente entre la distancia mínima y la distancia máxima, así:

$$C = \frac{d_{min}}{d_{max}} \tag{3.11}$$

Por lo tanto, un valor alto indica un mejor rendimiento del algoritmo de clustering.

3.5.4. Índice de Sorensen- Dice

El coeficiente de Sørensen-Dice, fue desarrollado por Thorvald Sørensen (Sørensen, 1948) y Lee Raymond Dice (Dice, 1945) y se utiliza para comparar la similitud de dos conjuntos.

Definición

De acuerdo a (Juárez Palavecino T.S, 2018) este coeficiente, al igual que el índice de Jaccard, da resultados en el rango $[0, 1]$ y se define de la siguiente manera para medir la similitud entre dos conjuntos:

$$SD(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{3.12}$$

$|A|$ y $|B|$ son las muestras respectivamente, y C es el conjunto de elementos de la intersección de las muestras A y B; $SD(A, B)$ es el cociente de similitud.

3.5.5. Indice de Jaccard

El índice de Jaccard, también conocido como Intersección sobre la Unión y el coeficiente de similitud de Jaccard (originalmente dado el nombre francés coeficiente de comunidad de Paul Jaccard), es un estadístico utilizado para medir la similitud y la diversidad de conjuntos de muestras. El coeficiente de Jaccard mide la similitud entre conjuntos de muestras finitas y se define como el tamaño de la intersección dividido por el tamaño de la unión de los conjuntos de muestras:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{3.13}$$

Según (Juárez Palavecino T.S, 2018) el índice Jaccard toma valores entre 0 (nada similar) y 1 (idéntico). Nótese que si A y B son conjuntos vacíos, $J(A, B) = 1$.

$$0 \leq J(A, B) \leq 1$$

Definición

El índice de Jaccard está definido como la intersección sobre la unión de los conjuntos y es una medida de la similitud entre ambos. Es decir, es la división entre el número de elementos en común que tienen los dos conjuntos sobre el número de elementos únicos que tiene la unión de ambos conjuntos:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.14}$$

Por su parte, la distancia de Jaccard es el resultado de restarle a 1 el valor de la similitud.

$$d_J(A, B) = 1 - J(A, B) \tag{3.15}$$

3.5.6. Calinski-Harabasz Index

(Ethen, 2015) La evaluación de Calinski Harabasz es un objeto que consiste en datos de muestra, datos de agrupación y valores de criterio de Calinski-Harabasz utilizados para evaluar el número óptimo de agrupaciones.

Se define el total de cuadrados de la siguiente manera:

$$\sum_{i}^{k} \sum_{x \in C_i} \|x - m_i\|^2 \tag{3.16}$$

Donde k denota el número de grupos, x es el punto de datos, C_i es el ***i-ésimo*** grupo, m_i es el centroide del grupo i, y $\|x - m_i\|$ es la norma $L2$ (distancia euclidiana) entre los dos vectores.

el centroide se obtiene tomando el valor medio de todos los puntos en ese grupo) Y el total dentro de la suma de cuadrados, es la suma de la suma interna de cuadrados de todos los grupos. Para esta medida, disminuirá en tanto que aumente el número de grupos, porq el cálculo de la fórmula se puede dividir en dos partes pequeñas. La suma de cuadrados dentro de un solo grupo es la distancia al cuadrado de cada punto en el grupo desde el centroide de ese grupoue cada grupo será más pequeño y más ajustado. Entonces, lo que esperamos es que la medida siga disminuyendo hasta el número de clúster óptimo, y la disminución comenzará a nivelarse después de eso. Si miramos la figura de ejemplo (3.4), al mirar hacia atrás en la trama de la derecha, hay un "codo" con 4 clústeres, donde la magnitud de del número de clústeres comienza a disminuir. Sin embargo, este "codo" a veces puede ser difícil de ver y puede ser bastante subjetivo.

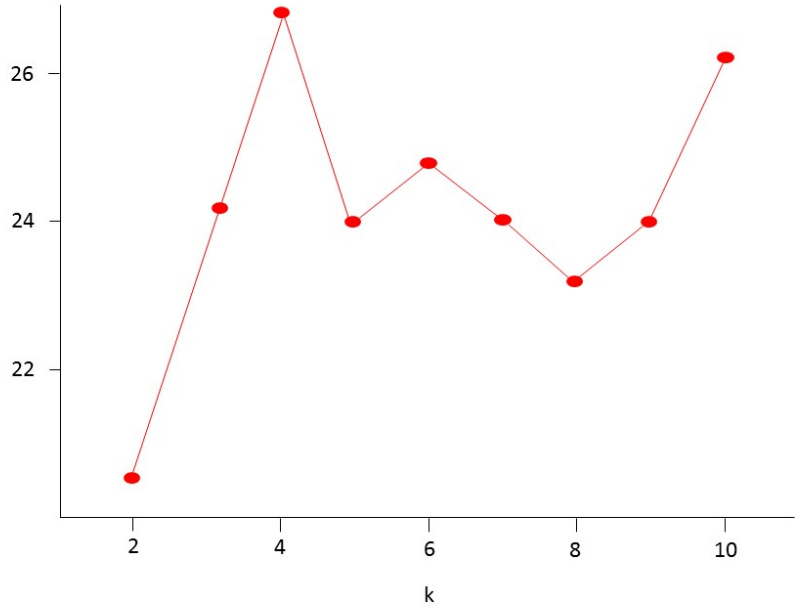

Figura 3.4: Indice CH

Definición

Sea k el número de grupos, y N es el número total de observaciones (puntos de datos), SSW es la varianza general dentro del grupo (equivalente al total dentro de la suma de cuadrados calculada anteriormente), SSB es la varianza global entre grupos.

$$CH = \frac{SS_B}{SS_W} \times \frac{N-k}{k-1} \tag{3.17}$$

Se puede calcular SSB utilizando la suma total de cuadrados (TSS) menos SSW. Para un conjunto de datos dado, la suma total de cuadrados (TSS) es la distancia al cuadrado de todos los puntos de datos desde el centroide del conjunto de datos, esta medida es independiente del número de conglomerados.

Para mayor claridad, SSB mide la varianza de todos los centroides del grupo desde el gran centroide del conjunto de datos (un valor SSB grande significa que el centroide de cada grupo se extenderá y no están demasiado cerca el uno del otro), y dado que ya sabemos SSW seguirá disminuyendo a medida que aumenta el número de clústeres. Por lo tanto, para el Índice Calinski-Harabasz, la proporción de $\frac{SSB}{SSW}$ debería ser la mayor que en el tamaño de agrupamiento óptimo.

En la figura (3.4), se puede ver claramente que esta medida es la más grande en el tamaño del cluster 4. Una cosa importante para usar esta medida es que a veces alcanza el nivel óptimo en el grupo 2, sin embargo , agrupar el punto de datos en 2 grupos puede no ser ideal, cuando esto suceda, la medida máxima local (la medida caerá y volverá a aumentar) debería ser su nú-

mero de grupo ideal.

El Índice de Calinski-Harabasz y la forma directa de ver el dendograma sugieren un tamaño de grupo de 4. Pero esto probablemente no sea cierto para cada conjunto de datos. Cuando eso sucede, depende del investigador decidir la heurísica adecuada para calcular el tamaño del clúster que sea más razonable.

3.5.7. Ball-Hall Index

Hay muchos ejemplos de índices de validez interna basados en la compacidad dentro de un grupo y la separación entre grupos. Los índices basados en la suma de cuadrados se basan en valores de suma de cuadrados dentro del grupo (SSW) y / o suma de cuadrados entre grupos (SSB), y el indice de Ball-Hall (BH) es uno de ellos.

Definición

(Ball y Hall, 1965) *sugirieron que la distancia promedio de los elementos a sus respectivos centroides de conglomerados podría servir como una medida útil del número de conglomerados en los datos. En la situación actual, se utiliza la mayor diferencia entre niveles para indicar la solución óptima. Se calcula usando la siguiente ecuación:*

$$BH_k = \frac{SSW_k}{k} \tag{3.18}$$

donde k es el número de grupos y SSW es suma de cuadrados dentro del grupo.

Por lo tanto se deduce, excepto que $SSW_{k+1} \leq SSW_k \forall k > 1$, que el índice BH disminuye a medida que k aumenta. Por lo tanto, en general, no se exceptúa que BH_k tenga minimizador o maximizador local. Esto siempre es cierto para los algoritmos de agrupamiento incremental para los cuales BH_k disminuye de forma monótona. En este caso, se puede definir una tolerancia $\epsilon > 0$.

Si $BH_k - BH_{k+1} \leq \epsilon$ para algunos $k \geq 2$, entonces k puede ser aceptado como el número óptimo de grupos.

Capítulo IV
ALGORITMO Y APLICACIÓN

En mineria de datos, el análisis de perfilamientos de individos de matrices de información, ha generado gran interés en los últimos años y la importancia de evaluar estos conglomerados, aunque tenemos gran información en la literatura científica, en la realidad empresarial se denota una falta de comparación entre algoritmos de segmentación y el contraste de los resultados teniendo en cuenta los índices de validación, debido a las siguientes razones:

- Desconocimiento en la composición de las matrices de información (pagina 11).
- Desconocimiento en las restricciones de los algoritmos (capítulo 2)
- Desconocimiento en las evaluaciones de perfilamientos a través de los índices de validación (capítulo 3)
- Costos computacionales

Por lo tanto, en este capítulo, se sugiere un algoritmo que permite la interacción de los resultados de varios métodos de segmentación para datos no supervisados comparando los indicadores de validación entre ellos.

Este algoritmo está enfocado principalmente en métodos de partición y/o difusos por las siguientes dos razones:

- En la actualidad son los más utilizados por su eficacia y simpleza de ejecución por lo tanto, su comparación debe ser tenida en cuenta cada vez que se desea investigar perfilamientos de individuos con estos algoritmos para datos no supervisados.
- Debido a la deficiencia que hay en la selección del número de clúster óptimo a seleccionar, el algoritmo permitirá no solo evaluar los resultados de cada método por clúster, sino que también permite evaluar los resultados entre números de clústeres k_i.

En línea con lo anterior, se presentará un caso práctico, donde se aplicará el algoritmo propuesto y evaluarán los resultados del perfilamiento de los individuos de estudio.

4.1. Algoritmo de interacción

El algoritmo de interacción consta de los siguientes pasos que se ilustrarán en la siguiente secuencia:

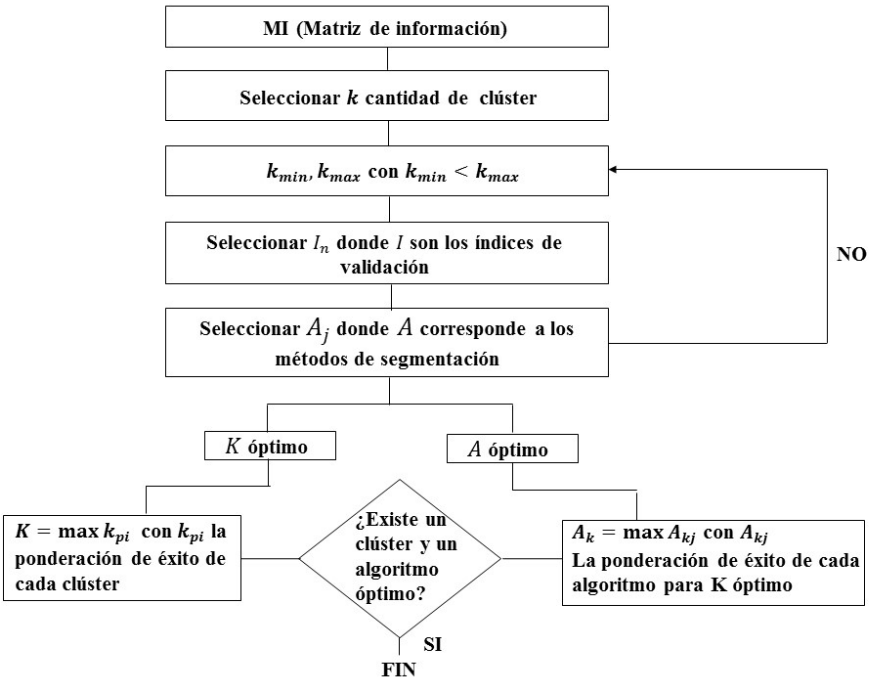

Figura 4.1: Algoritmo de interacción

4.1.1. Identificación

En primer lugar el usuario y/o investigador debe seleccionar k_{min} y k_{max} cumpliendo la condición de que $k_{min} \leq k_{max}$, donde k_{min} es el mínimo y k_{max} es el máximo de conglomerados a evaluar respectivamente.

4.1.2. Cuadro característico M

Luego de seleccionar los agrupamientos, se realiza un análisis descriptivo de los métodos A_j a evaluar teniendo en cuenta las siguientes características (Halkidi, Batistakis, y Vazirgiannis, 2001):

a) Tipo de datos: Identificar para que clase de datos aplica el algoritmo (numérico, categórico, mixto).

b) Complejidad: n es el número de puntos en el conjunto de datos y k el número de grupos definidos.

c) Geometría: Composición geométrica de los datos (convexos, no convexos).

d) Parámetros de Entrada: Aparte del número de clúster, indentificar que otro parámetro de entrada tienen los algoritmos seleccionados.

e) Criterio de segmentación: Panorama general de los criterios que utiliza cada método para perfilar los individuos de estudio.

4.1.3. Cuadro característico I

Simultáneamente a la sección anterior se debe identificar y seleccionar los índices de validación I_n con los que desee trabajar , teniendo en cuenta los siguientes criterios:

a) Enfoque: Qué tipo de validación es, interna, externa o relativa (se sugiere al lector validar los resultados de los conglomerados con los mencionados en el capítulo 3 ya que están entre los más utilizados y eficaces en la literatura y campo).

b) Criterio de validez: Panorama general de los criterios que utiliza cada índice para evaluar los resultados de cada algoritmo de segmentación.

c) Interpretación: Como se debe interpretar el resultado que arroja el índice de validez.

4.1.4. Ejecución M

Al tener los métodos seleccionados se ejecuta cada algoritmo a la matriz de información, que es objeto de investigación. Esta "corrida" se ejecuta en dos grupos de iteraciones:

1. Iteración interna: Se ejecuta las iteraciones de cada algoritmo de segmentación para encontrar el perfilamiento óptimo de acuerdo a su criterio de segmentación.

2. Iteración por cada k_i: Iteraciones por cada clúster seleccionado k_i con $i = k_{min}, k_{min+1}, .., k_{max}$, teniendo en cuenta el item anterior, a través de una secuencia cíclica.

Estas iteraciones permiten obtener el resultado de segmentación R_{ij} teniendo en cuenta el *i-ésimo K* clúster para el *j-ésimo* algoritmo.

4.1.5. Ejecución I

Luego cada resultado de agrupación, por cada método, se evalúa la eficacia de sus conglomerados teniendo en cuenta el análisis por cada índice de validación I_n. Se pueden tener en cuenta todos lo índices que el investigador desee.

Con este análisis obtenemos V_{ijn} que es el resultado de la validez del *n-ésimo* índice para el *i-ésimo K* cluster con base en el *j- ésimo* algoritmo.

4.1.6. Interacción

Luego de la *Ejecución M* y de la *Ejecución I*, se generan don matrices que permiten ver la interacción de los resultados de los algoritmos por cada k_i clúster seleccionado y los resultados de cada algoritmo A_j contrastando cada índice de validación:

	k_1	k_2	...	k_i
I_1	A_{11}	A_{12}	...	A_{1i}
I_2	A_{21}	A_{22}	...	A_{2i}
...	...	...	...	...
I_I	A_{I1}	A_{I2}	...	A_{Ii}

Tabla 4.1: A óptimo por cada índice I_i para cada cada k_i clúster

	A_1	A_2	...	A_M
I_1	k_{11}	k_{12}	...	k_{1M}
I_2	k_{21}	k_{22}	...	k_{2M}
...	...	...	...	...
I_I	k_{I1}	k_{I2}	...	k_{IM}

Tabla 4.2: k óptimo por cada índice I_i para cada método A_j

4.1.7. Interpretación

Luego de ejecutar el algoritmo de interacción, los resultados se pueden resumir y analizar en dos enfoques:

1. ***k* clúster no definido:** Si el investigador y/o usuario no tiene restricciones en cuanto al número de conglomerados que se deseen para perfilar a los individuos de estudio, el algoritmo anteriormente presentado le proporcionará el panorama para definir lo siguiente:

   a) k óptimo local: De acuerdo a la interacción de la validación de los conglomerados se puede seleccionar el número de agrupamientos optimo como:

   $$k = \max kp_i \ \ \forall\, i = k_{min}, ..., k_{max-1}, k_{max}. \tag{4.1}$$

   Donde kp_i es la ponderación de éxito de cada ***i-ésimo*** clúster seleccionado. Esta ponderación local se calcula teniendo como base la tabla (4.1):

   $$kp_i = \frac{1}{M * I} \sum_{j=1}^{M} \sum_{n=1}^{I} C_{ijn}, \tag{4.2}$$

   Donde el ***j-ésimo***, es algoritmo de segmentación y ***n-ésimo***, es el índice de validación para que C_{ijn} las siguientes dos soluciones:

   $$C_{ijn} = \begin{cases} 1 & \text{Si } k_i \text{ es considerado el clúster óptimo en el algoritmo } j \text{ por él indice } n \\ 0 & \text{En caso contrario} \end{cases}$$

   b) Método de segmetanción óptimo local: Al tener k definido por la ecucación (4.1),

se selecciona el algoritmo más indicado mediante la siguiente ponderación:

$$A_k = \text{máx}\, A_{k_j} \;\; \forall\, j = 1, ..., M, \tag{4.3}$$

Donde A_{k_j} es la ponderación de éxito de cada *j-ésimo* algoritmo seleccionado para el clúster óptimo ya encontrado. Esta ponderación se calcula teniendo como base la tabla (4.2):

$$A_{k_j} = \frac{1}{I} \sum_{n=1}^{I} S_{kjn}. \tag{4.4}$$

Donde el *n-ésimo* índice de validación, me indica si el índice j para el clúster óptimo k, el algoritmo es el índicado para segmentar y/o perfilar los índividuos de estudio de la matríz de información. S_{kjn} tiene dos soluciones:

$$S_{ijn} = \begin{cases} 1 & \text{Si } A_j \text{ es el algoritmo } j \text{ óptimo para él indice } n \\ 0 & \text{En caso contrario} \end{cases}$$

2. ***k* clúster predefinido:** Si el investigador y/o usuario ya tienen predefinido el número de grupos a perfilar debido a temas externos como comportamientos del campo de estudio, que exigen un número fijo de conglomerados entre ortros, entonces el algoritmo de interacción le proporciona, mediante los índices de validación, el algoritmo que mejor perfila de acuerdo a los índices de validación teniendo en cuenta la ponderación de la interacción (4.3).

   Si k por algún motivo del investigador llega a modificarse, se recomienda aplicar nuevamente la ecuación (4.3) ya que no necesiaramente el método de segmentación utilizado sea el mejor, es decir, los algoritmos de segmentación óptimos pueden variar dependiendo del *k-ésimo* clúster predefinido.

Aunque el algoritmo de interacción está pensado especialmente para datos no supervisados con algoritmos de segmentación de partición, se puede implementar con datos supervisados y/o métodos de "clasificación" teniendo los análisis pertinentes de los pasos (4.1.2) y (4.1.3) respectivamente.

4.2. Aplicación

(Fajardo Moreno, 2020) plantea en su tesis doctoral, un modelo de investigación para determinar el grado de capacidad dinámica de absorción (CDA) en gerencia de proyectos en organizaciones en Colombia. Los subconjuntos, dimensiones y determinantes permiten a las organizaciones encontrar sus fortalezas y debilidades, para posteriormente tener una formulación de sus planes de mejoramiento. (Fajardo Moreno, 2020) selecciona una muestra de empresas, donde se les calificó ciertas variables para poder determinar el CDA, generando así una matriz muestral (mm) .

En línea con lo anterior, en este trabajo de investigación se utilizará dicha matriz muestral (mm), que tiene información dividida en dos aspectos:

- Variables categóricas descriptivas de las organizaciones seleccionadas para el estudio donde se tiene:
  1. Tipo de sociedad
  2. Actividad de la empresa
  3. Rango de ingresos Totales de la empresa
  4. Rango No de empleados
- Variables numéricas de acuerdo a una calificación determinada para calcular el CDA.

En este trabajo de investigación se presentará un análisis de segmentación teniendo en cuenta las variables númericas para identificar, de acuerdo a las variables calificativas, si existen patrones entre las empresas que me permitan ver la relación dentro y entre las empresas de la muestra, utilizando el algoritmo de "interacción" propuesto en este capítulo.

El presente trabajo se enfocarán más los resultados de la interacción del algoritmo que la explicación como tal de la interpretación del comportamiento de las empresas de la matriz de información "mm".

Se presentará paso a paso los resultados del algoritmo de "interacción" basados en el lenguaje de programación R.

4.2.1. Identificación

Ya que no se tiene una restricción del número de conglomerados que se debe tener, se selecciona varios k_i para analizar los resultados:

- $k_{min} = 3$. Por lo general, en la práctica, se desea perfilar individuos de estudio en conglomerados mayor o igual a 3 grupos ya que con solo 2 sería dificil determinar resultados concluyentes.
- $k_{min} = 10$. Pueden evaluarse e interactuar todos los grupos que se deseen pero mas de 10 segmentaciones no se consideran necesarias en esta investigación. Además el interactuar con mas grupos puede generar un mayor costo computacional.

4.2.2. Cuadro característico M

Debido a que los datos de la matriz mxm son "no supervisados", y de tipo numérico en las variables de estudio, se presenta el cuadro característico de los métodos vistos a profundidad en el capítulo 2. (Halkidi y cols., 2001).

Algoritmo	Tipo	Complejidad	Geometría	P. E.
K-medias	numérico	$O(n)$	Formas no convexas	N^o de clusteres
K-medoides	numérico	$O(k(n-k)^2$	Formas no convexas	N^o de clusteres
CLARA	numérico	$O(k(40+k)^2+k(n-k))$	Formas no convexas	N^o de clusteres
FCM	numérico	$O(n)$	Formas no convexas	N^o de clusteres

Algoritmo	C. de segmentación
K-medias	Ecu 2.0
K-medoides	$\text{mín}(T_{ih})$, Ecu 2.3
CLARA	$\text{mín}(T_{ih})$, Ecu 2.3 (K_{jih}: El costo de reemplazar el centro i por h en lo que respecta a O_j)
FCM	Ecuación 2.8

Tabla 4.3: Cuadro Caracterísito M

4.2.3. Cuadro característico I

Ya identificados los algoritmos de segmentación a utilizar, Se evaluarán los resultados con índices de validación internos. Específicamente los profundizados en el capítulo 3, donde tenemos el siguiente resumen:

Indice	Enfoque	C. Validez	Interpretación
Silueta	Interno	$S = \frac{1}{n}\sum s(i)$	$S \approx 1$
DB	Interno	$DB = \frac{1}{N}\sum_{i=1}^{N} D_i$	$min(DB)$
Dunn	Interno	$DI = \frac{\text{mín}\,\delta(C_i,C_j)}{\text{máx}\,\nabla_{ik}}$	$max(DI)$
CH	Interno	$CH = \frac{SS_B}{SS_W}\ x\ \frac{N-k}{k-1}$	$max(CH)$
Ball-Hall	Interno	$BH_k = \frac{SSW_k}{k}$	$min(BH)$

Tabla 4.4: Cuadro Característico I

4.2.4. Ejecución M

De acuerdo a la figura (1.1) del proceso KDD, antes de ejecutar este paso (minería de datos) es necesario preparar (Selección-procesamiento-transformación) la matriz mm para no generar conclusiones que tengan insuficiente argumento real del comportamiento de los individuos de estudio. En la literatura científica se encuentran mucha información sobre este proceso, por lo tanto en esta investigación se omité la presentación de la preparación de los datos. Luego se procede a la ejecución de los algoritmos de segmentación presentados en el paso 4.2.2.

Para esta aplicación la ejecución de los algoritmos se basó en el entorno de software estadístico R como ya se mencionaron anteriormente para generar el conjunto de agrupaciones k_i con $k_{min} \leq k_i \leq k_{max}$.

Para que los datos puedan replicarse es necesario plantar una semilla. Para ejecutar las iteraciones internas y por cada k_i para cada algoritmo estudiado, se utiliza el comando for que me permite hacer este proceso simultaneamente de manera cíclica.

4.2.5. Ejecución I

Luego de ejecutar cada algoritmo de segmentación a la matriz mm, se evalúan los resultados del comportamiento de perfilamiento que sugiere los metodos. Ya que se está utilizando el entorno R, de manera simultánea cada vez que se ejecute el algoritmo de acuerdo al comando for, se puede evaluar los resultados de cada conjunto k_i de agrupamientos de manera inmediata. En este caso los índices de validación utilizados fueron los estudiados en el capítulo 3.

Al Ejecutar este paso, para cada algoritmo teniendo en cuenta cada k_i se tiene el siguiente cuadro resumen de resultados:

Clúster	índices	K-media	K-medoides	CLARA	FCM
3	davies bouldin	1.62	1.44	1.62	1.64
3	dunn	0.13	0.15	0.13	0.14
3	silhouette	0.23	0.24	0.23	0.23
3	Calinski Harabasz	72.49	66.85	72.49	72.19
3	ball hall	6.17	5.76	6.17	6.21
4	davies bouldin	1.87	1.72	1.87	1.95
4	dunn	0.14	0.14	0.14	0.14
4	silhouette	0.17	0.18	0.17	0.16
4	Calinski Harabasz	56.27	53.73	56.27	55.22
4	ball hall	5.65	5.24	5.65	5.72
5	davies bouldin	1.81	1.94	1.81	2.3
5	dunn	0.15	0.15	0.15	0.16
5	silhouette	0.15	0.15	0.15	0.14
5	Calinski Harabasz	47.74	45.2	47.74	44.78
5	ball hall	5.46	5.25	5.46	5.51
6	davies bouldin	1.83	1.9	1.83	2.36
6	dunn	0.18	0.15	0.18	0.12
6	silhouette	0.14	0.14	0.14	0.12
6	Calinski Harabasz	41.79	38.44	41.79	38.75
6	ball hall	5.17	4.94	5.17	5.33
7	davies bouldin	1.69	1.99	1.69	2.72
7	dunn	0.15	0.18	0.15	0.14
7	silhouette	0.16	0.13	0.16	0.09
7	Calinski Harabasz	38.14	34.45	38.14	32.58
7	ball hall	4.91	5.1	4.91	5.4
...	...	...	...	...	...
10	davies bouldin	1.68	1.89	1.68	3.37
10	dunn	0.15	0.17	0.15	0.12
10	silhouette	0.13	0.1	0.13	0.04
10	Calinski Harabasz	30.77	27.77	30.77	24.79
10	ball hall	4.85	4.68	4.85	5.16

Tabla 4.5: Cuadro resultados I

4.2.6. Interacción

Al tener como primera interacción el cuadro resumen I (Tabla 4.5), se puede identificar el mejor algoritmo A_{ni} de segmentación por cada índice I_n de validación teniendo en cuenta cada conjunto de clusteres k_i.

índices	Kmeans	K-medoides	FCM	CLARA
DB	9	3	3	9
Dunn	6	7	5	6
Silhouette	3	3	3	3
CH	3	3	3	3
Ball-Hall	9	9	9	9

Tabla 4.6: Tabla resumen de interacción K óptimo

Cluster	Ball-Hall	CH	DB	Dunn	Silhouette
3	K-medoides	Kmeans	K-medoides	K-medoides	K-medoides
4	K-medoides	Kmeans	K-medoides	Kmeans	K-medoides
5	K-medoides	Kmeans	Kmeans	FCM	K-medoides
6	K-medoides	Kmeans	Kmeans	Kmeans	Kmeans
7	Kmeans	Kmeans	Kmeans	K-medoides	Kmeans
8	Kmeans	Kmeans	Kmeans	K-medoides	Kmeans
9	Kmeans	Kmeans	Kmeans	K-medoides	Kmeans
10	K-medoides	Kmeans	Kmeans	K-medoides	Kmeans

Tabla 4.7: Tabla resumen de interacción A óptimo

Estas matrices generan un panorama general de la "interacción" donde se puede concluir de manera visual las siguientes conclusiones:

- Para dos índices de validez, en todos los algoritmos se concluye que 3 conglomerados es el nivel óptimo de segmentación para entender el comportamiento de los índividuos de estudio.

- En su gran mayoría los algoritmos *K-medoides* y *K-means* muestran un mejor perfilamiento de los individuos de estudio en casi todos los conglomerados. Sólo en el caso de que sean 5 grupos y para el índice de validación Dunn se ve que los puntos frontera de los segmentos no están bien definidos y por lo tanto sugiere que el algoritmo *FC-means* sería el mejor método para agrupar los datos. Esta tabla fue generada teniendo en cuenta el código en R presentado después de la sección siguiente, donde se evalúa cada algoritmo de segmentación por cada índice de validación.

4.2.7. Interpretación

Teniendo la tabla (4.6) se puede generar a través de un conteo ponderado cual es el méjor k y posteriormente basado en la tabla (4.7) luego identificar A_k:

1. K óptimo local: teniendo en cuenta las ecuaciones (4.1) y (4.2) aplicadas a la tabla resumen de interacción (4.6) se tiene lo siguiente:

Cluster	K
3	0.5
4	0
5	0
6	0.1
7	0.05
8	0.05
9	0.3
10	0

Tabla 4.8: Tabla k óptimo

Por lo tanto $K = 0{,}5$ corresponde al $k_i = 3$, es decir, el $50\,\%$ de éxitos que tuvo el clúster $k_i = 3$ de ser mejor seleccionado entre la relación índice y algoritmo.

2. Algoritmo óptimo A_k: ya que se sabe el número de clúster óptimo, se procede a identificar cual es el mejor algoritmo que puede perfilar mejor a los individuos segmentandolos en dichos grupos.

Clúster	Algoritmo	K
3	K-medoides	0.8
4	K-medoides	0.6
5	Kmeans	0.4
5	K-medoides	0.4
6	Kmeans	0.8
7	Kmeans	0.8
8	Kmeans	0.8
9	Kmeans	0.8
10	Kmeans	0

Tabla 4.9: Tabla k óptimo

El $80\,\%$ de los índices indican que el mejor algoritmo para segmentar los datos en 3 grupos de acuerdo al resultado en las variables numéricas es el $k - medoides$

Por lo tanto con la data recolectada en la matriz mm se puede concluir que para un óptimo perfilamiento de individuos teniendo en cuenta los algoritmos de segmentación estudiados lo mas recomendable es segmentar los individuos en tres grupos utilizando el algoritmo de k-medoides.

Si el investigador llega a la conclusión de que tres grupos no son suficientes para segmentar mm, se sugiere entonces segmentar la matriz en 9 grupos utilizando el algoritmo de $k - means$

Capítulo V
CONCLUSIONES

El algoritmo de interacción, como su nombre lo índica, muestra un panorama general de cada resultado permitiendo una interacción entre los índices, algoritmos y conjunto de segmentos, con el fin de identificar la necesidad a los siguientes panoramas:

- Interacción en la validez de los perfiles por cada conjunto de segmentos k_i ya que se ve la ponderación de convergencia de los índices de validación teniendo en cuenta los algoritmos de segmentación de acuerdo a cada k_i. Para el caso aplicado de la matriz mm, recomienda que las empresas se pueden clasificar de manera óptima en 3 grupos de acuerdo a los resultados de las variables calificativas presentadas por (Fajardo Moreno, 2020).

- Este algoritmo también sugiere que si 3 grupos no son suficientes para satisfacer las necesidades del investigador, no necesariamente el algoritmo de segmentación tenga que ser el mismo para otro conjunto de segmentación ya que para el caso práctico, el algoritmo de interacción recomienda que para $k_i = 3\ y\ 4$ es mas efectivo utilizar el método K-medoides, miéntras que para $k_i = 6, 7, 8, 9, 10$ es mas efectivo el perfilamiento segmentando con el algoritmo de $k - means$. Hay casos como en $k_i = 5$, donde se puede utilizar cualquiera de los dos algoritmos sugeridos ya que la validez de los resultados no arroja un algoritmo predilecto.

Aunque este algoritmo está diseñado principalmente para bases de datos "no supervisados" se puede adaptar para datos supervisados, variables categóricas y/o mixtas (numéricas y categóricas) teniendo como base, en el análisis preliminar, las características de las variables de la matríz de información que es objeto de estudio.

Aunque el algoritmo de interacción arroja resultados óptimos, tiene un costo cumpatacional grande en los siguientes casos:

- La matriz de información tienen dimensiones muy grandes, debido a las iteraciones que se tienen que hacer por cada algoritmo de segmentación.

- Si K_i es mayor a 10, evaluar conjuntos de agrupamientos puede generar demoras en la salida de los resultados.

Por lo tanto se sugiere al investigador continuar con esta investigación en entornos de software estadísticos para reducir estos costos computaciones que pueden impedir el uso óptimo de este algoritmo.

Capítulo VI
ANEXOS

Se anexa El codigo completo de R utilizado incluido los pasos “interacción” e “interpretación” teniendo presente las siguientes librerias

1. dplyr: tratamiento de data frame
2. cluster: Aplicar los algoritmos *CLARA y K-medoides*
3. ppclust: Aplicar el algoritmo *Fc-means*
4. clusterCrit: Aplicar índices de validación

```r
set.seed(222)
vector_final<-NULL
for (i in kmin:kmax) {
  #--------------------K MEDIAS--------------------------------#
  # Algoritmo
  kmedias <- kmeans(DATOS1, centers = i, iter.max = 1000, nstart = 10,
algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"),
trace=FALSE)
  print(kmedias)
  # Validación
  val_indx_Kmedias <- intCriteria(DATOS1,kmedias$cluster,indices)
  Kmeanss<-c(val_indx_Kmedias[1],val_indx_Kmedias[2],val_indx_Kmedias[3],
val_idx_Kmedias[4],val_indx_Kmedias[5])

  #-------------------K MEDIODES----------------------#
  # Algoritmo
  Kmedoid<- pam(DATOS1, k=i)
  print(Kmedoid)
  # Validación
  val_indx_Kmedoid <- intCriteria(DATOS1,Kmedoid$clustering ,indices)
  PAM<-c(val_indx_Kmedoid[1],val_indx_Kmedoid[2],val_indx_Kmedoid[3],
val_indx_Kmedoid[4],val_indx_Kmedoid[5])

  #--------------------FUZZY C MEANS-------------#
  # Algoritmo
  fcmDATOS1<- fcm(DATOS1, centers=i)
  print(fcmDATOS1)
  # Validación
  val_indx_fcm <- intCriteria(DATOS1,fcmDATOS1$cluster ,indices)
  Fcmeans<-c(val_indx_fcm[1],val_indx_fcm[2],val_indx_fcm[3],
val_indx_fcm[4],val_indx_fcm[5])

  #--------------------CLARA--------------#
  # Algoritmo
  Clara<- clara(DATOS1,k=i,metric = c("euclidean", "manhattan",
"jaccard"))
  print(Clara)
  # Validación
  val_indx_Clara <- intCriteria(DATOS1,Clara$clustering ,indices)
  CLARA<-c(val_indx_Clara[1],val_indx_Clara[2],val_indx_Clara[3],
val_indx_Clara[4],val_indx_Clara[5])

  #-------------------Cuadro Resumen---------#
  vecto_fin<-rbind(indices,Kmeanss,PAM,Fcmeans,CLARA)
  print(vecto_fin)
  vecto_fin<-t(vecto_fin)
  vecto_fin<-as.data.frame(vecto_fin)
  vecto_fin$Cluster<-i
  vector_final<-rbind(vector_final,vecto_fin)
  }
```

```r
#-----------------------------------------------------------#
#--------MEJOR ALGORITMO POR CADA ÍNDICE DE VALIDACIÓN-------#
#-----------------------------------------------------------#

#--Evaluando los algoritmos con el índice davies_bouldin

vector_final$resumen_DB<-if_else(vector_final$indices=="davies_bouldin" &
vector_final$Kmeanss<=vector_final$PAM &
vector_final$Kmeanss<=vector_final$Fcmeans &
vector_final$Kmeanss<=vector_final$CLARA,"Kmeans",
if_else(vector_final$indices=="davies_bouldin" &
vector_final$PAM<vector_final$Kmeanss & vector_final$PAM<vector_final$Fcmeans  &
vector_final$PAM<vector_final$CLARA,"PAM",
if_else(vector_final$indices=="davies_bouldin" &
vector_final$CLARA<vector_final$Kmeanss &
vector_final$CLARA<vector_final$Fcmeans & vector_final$CLARA<=vector_final$PAM,
"CLARA","FCM")))

vector_final$resumen_DB<-if_else(vector_final$indices!="davies_bouldin", "",
vector_final$resumen_DB)

#--Evaluando los algoritmos con el índice dunn

vector_final$resumen_Dunn<-if_else(vector_final$indices=="dunn" &
vector_final$Kmeanss>=vector_final$PAM &
vector_final$Kmeanss>=vector_final$Fcmeans &
vector_final$Kmeanss>=vector_final$CLARA,"Kmeans",
if_else(vector_final$indices=="dunn" & vector_final$PAM>vector_final$Kmeanss &
vector_final$PAM>vector_final$Fcmeans &
vector_final$PAM>vector_final$CLARA,"PAM",if_else(vector_final$indices=="dunn" &
vector_final$CLARA>vector_final$Kmeanss &
vector_final$CLARA>vector_final$Fcmeans & vector_final$CLARA>=vector_final$PAM,
"CLARA","FCM")))

vector_final$resumen_Dunn<-if_else(vector_final$indices!="dunn","",
vector_final$resumen_Dunn)

#--Evaluando los algoritmos con el índice silhouette

vector_final$resumen_silhouette<-if_else(vector_final$indices=="silhouette" &
vector_final$Kmeanss>=vector_final$PAM &
vector_final$Kmeanss>=vector_final$Fcmeans &
vector_final$Kmeanss>=vector_final$CLARA,"Kmeans",
if_else(vector_final$indices=="silhouette" &
vector_final$PAM>vector_final$Kmeanss & vector_final$PAM>vector_final$Fcmeans &
vector_final$PAM>vector_final$CLARA,"PAM",
if_else(vector_final$indices=="silhouette" &
vector_final$CLARA>vector_final$Kmeanss &
vector_final$CLARA>vector_final$Fcmeans & vector_final$CLARA>=vector_final$PAM,
"CLARA","FCM")))

vector_final$resumen_silhouette<-if_else(vector_final$indices!="silhouette","",
vector_final$resumen_silhouette)
```

```r
#--Evaluando los algoritmos con el índice Calinski_Harabasz

vector_final$resumen_CH<-if_else(vector_final$indices=="Calinski_Harabasz" &
vector_final$Kmeanss>=vector_final$PAM &
vector_final$Kmeanss>=vector_final$Fcmeans &
vector_final$Kmeanss>=vector_final$CLARA,"Kmeans",
if_else(vector_final$indices=="Calinski_Harabasz" &
vector_final$PAM>vector_final$Kmeanss & vector_final$PAM>vector_final$Fcmeans &
vector_final$PAM>vector_final$CLARA,"PAM",
if_else(vector_final$indices=="Calinski_Harabasz" &
vector_final$CLARA>vector_final$Kmeanss &
vector_final$CLARA>vector_final$Fcmeans & vector_final$CLARA>=vector_final$PAM,
"CLARA","FCM")))

vector_final$resumen_CH<-if_else(vector_final$indices!="Calinski_Harabasz","",
vector_final$resumen_CH)

#--Evaluando los algoritmos con el índice ball_hall

vector_final$resumen_BH<-if_else(vector_final$indices=="ball_hall" &
vector_final$Kmeanss<=vector_final$PAM &
vector_final$Kmeanss<=vector_final$Fcmeans &
vector_final$Kmeanss<=vector_final$CLARA,"Kmeans",
if_else(vector_final$indices=="ball_hall" &
vector_final$PAM<vector_final$Kmeanss & vector_final$PAM<vector_final$Fcmeans  &
vector_final$PAM<vector_final$CLARA,"PAM",
if_else(vector_final$indices=="ball_hall" &
vector_final$CLARA<vector_final$Kmeanss &
vector_final$CLARA<vector_final$Fcmeans & vector_final$CLARA<=vector_final$PAM,
"CLARA","FCM")))

vector_final$resumen_BH<-if_else(vector_final$indices!="ball_hall","",
vector_final$resumen_BH)

#--Vector con la información completa
vector_final$Mejor_Alg<-paste0(vector_final$resumen_DB,
vector_final$resumen_Dunn,vector_final$resumen_silhouette,
vector_final$resumen_CH,vector_final$resumen_BH)
vector_final<-vector_final %>%
  dplyr::select(-resumen_DB,-resumen_Dunn,-resumen_silhouette,-resumen_CH,-
resumen_BH)

#-------------------------------------------------------------#
#--------MEJOR ALGORITMO POR CADA # DE CLUSTER---------------#
#-------------------------------------------------------------#

Mejor_Alg<-vector_final %>% mutate(contar=1) %>%  group_by(Cluster,Mejor_Alg)
%>% summarise(porcentaje_Mejor_Alg=sum(contar))
Mejor_Alg$porcentaje_Mejor_Alg<-Mejor_Alg$porcentaje_Mejor_Alg/length(indices)
Mejor_Alg1<-Mejor_Alg %>% group_by(Cluster) %>%
mutate(mejor_alg=max(porcentaje_Mejor_Alg))
#Resumen_Algoritmo
Mejor_Alg1<-Mejor_Alg1 %>% filter(mejor_alg==porcentaje_Mejor_Alg) %>%
dplyr::select(Cluster,Mejor_Alg,porcentaje_Mejor_Alg)
```

```r
#-------------------------------------------------------------#
#----------------MEJOR ALGORITMO DE SEGMENTACIÓN--------------#
#-------------------------------------------------------------#

mejor_cluster<-NULL
for (k in 1:length(indices)) {
  ind<-indices[k]
  vector_final1<-vector_final %>%
    dplyr::filter(vector_final$indices==ind)
  vector1<-vector_final1 %>%
    dplyr::filter(vector_final1$indices %in% c("davies_bouldin","ball_hall"))
%>%
    group_by(indices) %>%
mutate(km=min(Kmeanss),pam=min(PAM),FCM=min(Fcmeans),clara=min(CLARA))
  vector2<-vector_final1 %>%
    dplyr::filter(vector_final1$indices %in%
c("dunn","silhouette","Calinski_Harabasz")) %>%
    group_by(indices) %>%
mutate(km=max(Kmeanss),pam=max(PAM),FCM=max(Fcmeans),clara=max(CLARA))
  mejor_cluster<-rbind(mejor_cluster,vector1,vector2)
  }

mejor_cluster$clu_km<-if_else(mejor_cluster$Kmeanss==mejor_cluster$km,
as.numeric(mejor_cluster$Cluster),0)
mejor_cluster$clu_pam<-if_else(mejor_cluster$PAM==mejor_cluster$pam,
as.numeric(mejor_cluster$Cluster),0)
mejor_cluster$clu_fcm<-if_else(mejor_cluster$Fcmeans==mejor_cluster$FCM,
as.numeric(mejor_cluster$Cluster),0)
mejor_cluster$clu_Clara<-if_else(mejor_cluster$CLARA==mejor_cluster$clara,
as.numeric(mejor_cluster$Cluster),0)
mejor_cluster$mejor_clu<-(mejor_cluster$clu_km+mejor_cluster$clu_pam+
mejor_cluster$clu_fcm+mejor_cluster$clu_Clara)/mejor_cluster$Cluster
mejor_cluster1<-mejor_cluster %>%
  group_by(Cluster) %>%
  summarise(mejor_clu=sum(mejor_clu)/(M*length(indices))) %>%
  mutate(mj=max(mejor_clu))

#Resumen_Cluster de acuerdo a la ponderación

mejor_cluster2<-mejor_cluster1 %>%
  dplyr::filter(mejor_clu==mj) %>%
  dplyr::select(Cluster,mejor_clu)

#-------------------------------------------------------------------------#
#--------RESUMEN DE # DE CLUSTER Y MEJOR ALGORITUMO--------------------#
#-------------------------------------------------------------------------#

Mejor_clus_Alg<-mejor_cluster2 %>% left_join(Mejor_Alg1,by=c("Cluster"))
```

Referencias

Agrawal, R., y Srikant, R. (1995). Mining sequential patterns. En *Proceedings of the eleventh international conference on data engineering* (pp. 3–14).

Agrawal, R., Srikant, R., y cols. (1994). Fast algorithms for mining association rules. En *Proc. 20th int. conf. very large data bases, vldb* (Vol. 1215, pp. 487–499).

Ankerst, M., Breunig, M. M., Kriegel, H.-P., y Sander, J. (1999). Optics: ordering points to identify the clustering structure. *ACM Sigmod record*, *28*(2), 49–60.

Ball, G. H., y Hall, D. J. (1965). *Isodata, a novel method of data analysis and pattern classification* (Inf. Téc.). Stanford research inst Menlo Park CA.

Banchero, S. (2015). Bases de datos masivas. , 1–2.

Bellido, G. F. (2017).

Benzécri, J. (1984). Description des textes et analyse documentaire. *Cahiers de l'analyse des données*, *9*(2), 205–211.

Benzécri, J.-P. (1977). El análisis de correspondencias. *Cahiers de l'analyse des données*, *2*(2), 125–142.

Bernard, D. (2017). Clustering indices.

Berry, M., y Linoff, G. (1997). *Data mining techniques: For marketing, sales and marketing support.* wiley.

Chen, Han, y Prinn. (2006). Estimation of atmospheric methane emissions between 1996 and 2001 using a three-dimensional global chemical transport model. *Journal of Geophysical Research: Atmospheres*.

Dave, R. N. (1996). Validating fuzzy partitions obtained through c-shells clustering. *Pattern recognition letters*, *17*(6), 613–623.

Davies, D. L., y Bouldin, D. W. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*(2), 224–227.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, *26*(3), 297–302.

Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, *4*(1), 95–104.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., y cols. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. En *Kdd* (Vol. 96, pp. 226–231).

Ethen, L. (2015).

Fajardo Moreno, W. S. (2020). Modelo para la medición de la capacidad dinámica de absorción en gerencia de proyectos.

Fayyad, Piatetsky-Shapiro, G., y Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, *17*(3), 37–37.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., y cols. (1996). Knowledge discovery and data mining: Towards a unifying framework. En *Kdd* (Vol. 96, pp. 82–88).

Gath, I., y Geva, A. B. (1989). Unsupervised optimal fuzzy clustering. *IEEE Transactions on pattern analysis and machine intelligence*, *11*(7), 773–780.

Halkidi, M., Batistakis, Y., y Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of intelligent information systems*, *17*(2-3), 107–145.

Han, I., y Kamber, M. (2001). Data mining: Concepts and techniques. *Morgan Kaufinann*.

Hartigan, J. A. (1985). Statistical theory in clustering. *Journal of classification*, *2*(1), 63–76.

Hernández, J., Ramírez, M., y Ferri, C. (2005). Introducción a la minería de datos. editorial.

Hinneburg, A., Keim, D. A., y cols. (1998). An efficient approach to clustering in large multimedia databases with noise.

Juárez Palavecino T.S, D. D. (2018).

Krishnapuram, R., y Keller, J. M. (1996). The possibilistic c-means algorithm: insights and recommendations. *IEEE transactions on Fuzzy Systems*, *4*(3), 385–393.

Lebart, L. (1994). Sur les analyses statistiques de textes. *Journal de la société française de statistique*, *135*(1), 17–36.

MacQueen, J., y cols. (1967). Some methods for classification and analysis of multivariate observations. En *Proceedings of the fifth berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281–297).

Maria Halkidi, M. V., Yannis Batistakis. (2001). On clustering validation techniques.

McInnes, L., Healy, J., y Astels, S. (2017). hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, *2*(11), 205.

Ng, R. T., y Han, J. (1994). E cient and e ective clustering methods for spatial data mining. En *Proceedings of vldb* (pp. 144–155).

Ng, R. T., y Han, J. (2002). Clarans: A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, *14*(5), 1003–1016.

Pastrán, L. F., y Roa, N. J. (2015). Clasificación mediante k-modas para el caso de variables categóricas.

Petrovic, S. (2006). A comparison between the silhouette index and the davies-bouldin index in labelling ids clusters. En *Proceedings of the 11th nordic workshop of secure it systems* (pp. 53–64).

Qinpei Zhao, P. F., Mantao Xu. (2009). Adaptive and natural computing algorithms. , 1–2.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81–106.

Rangel, E. M., Hendrix, W., Agrawal, A., Liao, W.-k., y Choudhary, A. (2016). Agoras: A fast algorithm for estimating medoids in large datasets. *Procedia Computer Science*, *80*, 1159–1169.

Rehioui, H., Idrissi, A., Abourezq, M., y Zegrari, F. (2016). Denclue-im: A new approach for big data clustering. *Procedia Computer Science*, *83*, 560–567.

Rezaee, M. R., Lelieveldt, B. P., y Reiber, J. H. (1998). A new cluster validity index for the fuzzy c-mean. *Pattern recognition letters*, *19*(3-4), 237–246.

ROUSSEEUW, P. J., y KAUFMAN, L. (1987). Clustering by means of medoids.

Rueda, M., Moya, L., y Silva, M. (2011). Aplicación de técnicas estadísticas multivariadas en

perfilación y segmentación. *Universitas Scientiarum*, *16*. doi: 10.11144/javeriana.SC16-3.uoms

Russell, S., y Norvig, P. (2002). *Artificial intelligence a modern approach*. Prentice Hall.

Sattler, K.-U., y Dunemann, O. (2001). Sql database primitives for decision tree classifiers. En *Proceedings of the tenth international conference on information and knowledge management* (pp. 379–386).

Sørensen, T. J. (1948). *A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons*. I kommission hos E. Munksgaard.

Theodoridis, S., y Koutroumbas, K. (1999). Pattern recognition and neural networks. En *Advanced course on artificial intelligence* (pp. 169–195).

Timarán Pereira S.R, C. Z. S. H. T. A. y. A. P. J., Hernández Arteaga L. (2016). El proceso de descubrimiento de conocimiento en bases de datos.

Wang, M., Iyer, B., y Vitter, J. S. (1998). Scalable mining for classification rules in relational databases. En *Proceedings. ideas'98. international database engineering and applications symposium (cat. no. 98ex156)* (pp. 58–67).

Xie, B. G., X.L. (1991). A validity measure for fuzzy clustering. ieee transactions on pattern analysis and machine intelligence. , 841–846.

Yu, D., Liu, G., Guo, M., y Liu, X. (2018). An improved k-medoids algorithm based on step increasing and optimizing medoids. *Expert Systems with Applications*, *92*, 464–473.

Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, *8*(3), 338–353.

Zhang, T., Ramakrishnan, R., y Livny, M. (1996). Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, *25*(2), 103–114.